TOWARDS VISUAL LEARNING WITH ATTENTION MECHANISM

by

LEZI WANG

A dissertation submitted to the School of Graduate Studies Rutgers, The State University of New Jersey
In partial fulfillment of the requirements For the degree of Doctor of Philosophy Graduate Program in Computer Science
Written under the direction of Dimitris N. Metaxas
And approved by

_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________

New Brunswick, New Jersey
May 2021
ABSTRACT OF THE DISSERTATION

Towards Visual Learning with Attention Mechanism

by Lezi Wang

Dissertation Director:
Dimitris N. Metaxas

Tremendous interest in deep learning has emerged in the computer vision research community. The established deep convolutional neural networks (CNNs) have achieved astonishing results in various vision tasks, while there are still problems that need to be addressed. First of all, the CNN models are perceived as “black-box” with a lack of understanding of the internal function. Recently, the class-wise activation map is proposed to show a visual explanation of model attention, while it still lacks the way to utilize that explanation to guide the learning process. Additionally, the success of deep learning relies on supervised training the models on the large-scale data, which requires humans to create massive annotations.

In this dissertation, we address that attention mechanisms can play significant roles in dealing with the challenges mentioned above. First, despite class-wise attention mechanisms providing good localization for an individual class of interest when it comes to interpreting CNNs, these techniques produce attention maps with substantially overlapping responses among different classes, leading to visual confusion and the need for discriminative attention. We address this problem by means of a new framework that makes class-discriminative attention a principled part of the learning process. Second, to get rid of human annotations, we introduce the Co-Attention as a weak-supervision to generate the positive/negative training samples and a Contrastive Attention module to enhance the feature representations such that the comparative contrast between features of the positive and negative samples are
maximized. Third, we adopt the attention in feature space to bridge different vision tasks in a unified learning framework. Extensive experiments on vision benchmarks show the effectiveness of our approaches in terms of improved image classification and segmentation accuracy. Regards the applications, our proposed algorithms are applied to the unsupervised detection of highlighted segments in the videos, joint face detection and landmark localization, and reasoning about human facial behaviors in deception. Additionally, two new benchmarks are collected to support related studies and facilitate researches in the same direction.
ACKNOWLEDGMENTS

I would like to thank Professor Dimitris N. Metaxas for his kind support and guidance during the past years. I am deeply grateful for his insightful talks and encouragement, which have helped me tremendously. I would like to thank my other committee members: Prof. Konstantinos Michmizos, Prof. Georgios Moustakides, and Prof. Yiannis Aloimonos for their valuable advice and comments regarding this thesis. It is my great honor to have each of them in my committee and help me to improve my work.

At Rutgers, I am fortunate to have collaborated with Prof. Mubbasir Kapadia, Dr. Xiang Yu and Bo Liu. I am deeply grateful for their precious discussions and encouragement when I was still a junior graduate student. Also, I would like to thank all the labmates at Rutgers CBIM lab. Thanks for the memories and sharing the excitement and frustrations along this unforgettable journey.

I would like to sincerely thank my internship mentors at Siemens Vision Group and Netflix, Ziyang Wu, Kuan-chuan Peng and Srikrishna Karanam, and Dong Liu, for their tremendous support, patience and huge inspiration as academic role models to me.

Last but not least, I never forget to thank my families, for their unselfish support and enduring me all these years. Please allow me to express my gratitude, now and always.
DEDICATION

To my parents
# TABLE OF CONTENTS

Abstract ........................................................................................................ ii

Acknowledgments ......................................................................................... iv

Dedication ......................................................................................................... v

List of Tables ..................................................................................................... ix

List of Figures .................................................................................................. xi

1. Introduction .................................................................................................. 1
   1.1 Visual Explanation ..................................................................................... 2
   1.2 Incorporate Attention into Learning Process ............................................. 4
   1.3 Reduce Annotation Budget ....................................................................... 5
   1.4 Contribution of the Dissertation ............................................................... 6
   1.5 Outline of the Dissertation ....................................................................... 7

2. Learning with Attention Separability and Consistency ............................. 9
   2.1 Introduction ............................................................................................... 9
   2.2 Related Work ........................................................................................... 12
   2.3 Approach .................................................................................................. 13
      2.3.1 Channel-weighted Attention $A_{ch}$ .................................................... 15
      2.3.2 Attention Separation Loss $L_{AS}$ ....................................................... 17
      2.3.3 Attention Consistency Loss $L_{AC}$ .................................................... 19
2.3.4 Overall Framework ICASC ........................................... 19
2.4 Experiments ................................................................. 20
  2.4.1 Evaluating Class Discriminability ............................... 20
  2.4.2 Evaluating ICASC for Image Classification .................... 23
2.5 Conclusion ................................................................. 34

3. An Coupled Encoder-Decoder Network for Joint Face Detection and Land-
mark Localization ............................................................ 35
  3.1 Introduction ............................................................... 35
  3.2 Related Work .............................................................. 37
  3.3 Coupled Encoder-Decoder Network .................................... 40
    3.3.1 The Encoder-Decoder for Facial Response Map Prediction ... 41
    3.3.2 The Coupled Feature Map Cascade for Face Detection ........ 42
    3.3.3 Landmark Localization From Response Maps ................. 43
  3.4 Implementation Details .................................................. 44
    3.4.1 Network Architectures ............................................ 44
    3.4.2 Training ............................................................ 45
  3.5 Experiments ............................................................... 47
    3.5.1 Experimental Setup ............................................... 47
    3.5.2 Evaluation of Face Landmark Localization .................... 47
    3.5.3 Evaluation of Face Detection .................................... 49
    3.5.4 Qualitative Results .............................................. 52
  3.6 Conclusion ............................................................... 53

4. Learning Trailer Moments in Full-Length Movies with Co-Contrastive At-
tention ............................................................... 55
  4.1 Introduction .............................................................. 55
  4.2 Related Works ............................................................ 58
4.3 Approach .......................................................... 60
  4.3.1 Trailer Moment Detection Dataset ......................... 60
  4.3.2 CCANet for Trailer Moment Detection .................. 61
4.4 Experiment Results ............................................... 65
  4.4.1 Movie Key Moment Detection Results .................... 65
  4.4.2 Video Highlight Detection Results ....................... 69
  4.4.3 Understanding the Co-Contrastive Attention ............ 71
4.5 Conclusion ....................................................... 72

5. Attention-based Facial Behavior Analytics in Social Communication .... 74
  5.1 Introduction ................................................... 74
  5.2 Related Work ................................................ 75
  5.3 Methodology .................................................. 77
    5.3.1 Training a 3D convolutional Neural Network for Spy Detection ... 79
  5.4 Experiments ................................................... 80
    5.4.1 Dataset .................................................. 80
    5.4.2 Results of Spy Detection ............................... 83
    5.4.3 Attention and Deception Cues .......................... 85
  5.5 Conclusion ................................................... 86

6. Conclusion ....................................................... 88

References ............................................................. 90
LIST OF TABLES

2.1 Results on Pascal VOC 2012 segmentation validation set. .......................... 22
2.2 Experimental (exp.) settings used in this paper. VGG, RN-18, RN-50, and RN-101 denote VGG-19 [28], ResNet-18 [2], ResNet-50, and ResNet-101, respectively. We use the same parameters as the references in the last row unless otherwise specified, putting more details in the supplementary material. Acronyms: BNA: base network architecture; WD: weight decay; MOM: momentum; LR: initial learning rate; BS: batch size; OPM: optimizer; SGD: stochastic gradient descent [40]; CCA: cyclic cosine annealing [41]. .......................... 23
2.3 Ablation study on CUB-200-2011 (Δ=performance improvement; “Top-1”: top-1 accuracy (%)). ................................................................. 25
2.4 Image classification results on CIFAR-100. .................................................. 27
2.5 Results on Caltech-256. “Top-5”: top-5 accuracy (%). “RN-18”: ResNet-18. “N”: # of training images per class. We follow [23] to randomly select 30 or 60 training images per class. ................................................................. 27
2.6 Results on ILSVRC2012. ............................................................................. 27
2.7 Results on CUB-200-2011. “No Extra Anno.” means not using extra annotation (bounding box or part) in training. “1-Stage” means the training is done in one stage. ................................................................. 28
2.8 Image classification results on Pascal VOC 2012. ........................................ 29
2.9 Categorical and mean Average Precision (mAP) (%) for our PASCAL VOC 2012 image classification experiment. The highest scoring entry is shown in bold. .......................................................... 30

2.10 Comparing baseline, ICASC \textsubscript{Grad–CAM} and ICASC \textsubscript{Ach}. .......................... 31

4.1 The trailer moment detection results on TMDD. “Sup”, “PL” and “CoA” denote the different approaches, including fully-supervised, Pseudo-Label and our Co-Attention, with the basic 3D features [144]. The “Sup+CA”, “PL+CA” and “CCANet” denote that the shot features in “Sup”, “PL” and “CoA” are augmented with our proposed Contrastive Attention module. The terms “Act”, “Dra” and “ScF” refer to the movie categories, i.e., Action, Drama and Sci-Fi, and “Avg” indicates the “Average” result across categories. The subscripts “10”, “20” and “GL” indicate different evaluation metrics of Rank@10, Rank@20 and Rank@Global. .............................................. 67

4.2 Rank@Global of the proposed CoA approach with different feature encoding strategies. .................................................. 68

4.3 The highlight detection mAP on YouTube Highlight dataset. Avg. is the average mAP over all the domains. Our approach outperforms all the baselines. 71

4.4 The highlight detection top-5 mAP score on TVSum [119]. The ‘-’ means that mAP value is not provided in the original paper. .......................... 71

5.1 The Spy/Villager prediction accuracy reported on the two different dataset splitting. The training data is randomly sampled without attention knowledge. 84

5.2 The Spy/Villager prediction accuracy reported on the two different dataset splitting. The model is trained with the video frames selected according to the model attention. .......................... 84
LIST OF FIGURES

1.1 The class-specific attention highlights the important pixels for CNN recognizing specific categories. The middle indicates the input image. The leftmost depicts the attention for the “Person” category; and the rightmost is the attention for “Bird”. ................................................................. 2

1.2 The Grad-CAM of different feature layers for two categories. The top indicates the attention maps for the category “common iguana” and bottom is “coho”. For each category, the attentions from higher to lower layers are listed left to right. For the same category, the model attention shift cross the layers. ................................................................. 3

1.3 The model with discriminative attention can tell various object categories apart. Given an input image with ground truth label “boom box”, we show the category-specific attention maps for top-4 predictions: “headphone”, “video projector”, “boom box”, and “toaster”. The attention map and prediction score for ground truth class “boom box” are marked by red bounding box and bar respectively. The less discriminative attention from vanilla CNN highlights very similar pixels, confusing the model to recognize the image as “boom box”. The more separable attention provides the better model interpretability, where the model attends different crucial pixels across various categories, eliminating the confusion and leading to better model discriminability. ................................................................. 5
2.1 The baseline CNN attends to similar regions, i.e. the central areas, when it comes to the relevant pixels for classes “treadmill,” “basketball hoop,” and “tripod.” The CNN with our proposed framework is able to tell the three classes apart and has high confidence to classify the input as “basketball hoop.”

2.2 Grad-CAM [9] attention maps of the VGG-19 [21] top-5 predictions. Predictions with red-bounding boxes correspond to the ground-truth class. (a) Ground-truth class attention maps from the last layer (Conv5) have a large overlap with false positives (top-1 predictions). (b) Inner-layer attention maps (Conv4) are more separable than their last-layer counterparts.

2.3 The framework of Improving Classification with Attention Separation and Consistency (ICASC).

2.4 The Grad-CAM [9] attentions of different VGG-19 [21] feature layers for the ‘tench’ class. In both rows, the target is the fish while the model attention shifts across the layers.

2.5 The comparison of attention maps from different VGG-19 [21] layers. Ours has less attention shift than Grad-CAM [9]. In the marked areas, ours attends to the target objects, i.e. bird, while Grad-CAM [9] tends to highlight the background pixels.

2.6 The top row demonstrates higher visual confusion than the bottom row. The two attention maps in the top row have high responses localized at the bird’s head, while as shown in the bottom, the ground-truth class attention highlights the bird’s head, and the confusing class attention addresses the lower body.

2.7 Multi-class attention maps (‘bird’ and ‘person’).
2.8 Segmentation masks generated from attention maps by DeepLab [35] (best view in color, zoom in). From left to right: the Input Image, Ground Truth, Grad-CAM, Grad-CAM++ and ours. ........................................................... 22

2.9 The KS-Chart on the CUB-200-2011 testing set. “Ours” stands for ResNet-50 + $L_{AS}^{ln} + L_{AS}^{la} + L_{AC}$ in Table 2.3. ........................................................... 25

2.10 Qualitative results with CIFAR-100. We show top-5 predictions with classification scores given by ResNet-110 (top row) and ResNet-110 + ICASC$_{A_{ch}}$ (bottom row). ........................................................... 26

2.11 The attention overlap histograms. ACC: Top-1 classification accuracy. . . . 31

2.12 Improvements in top-1 predictions with our method (ResNet-18 + ICASC$_{A_{ch}}$) when compared to the baseline (ResNet-18). Top row: ResNet-18; bottom row: ResNet-18 + ICASC$_{A_{ch}}$. ........................................................... 32

2.13 Improvements in top-1 predictions with ResNet-110 + ICASC$_{A_{ch}}$ when compared to the baseline (ResNet-110). Top row: ResNet-110; bottom row: ResNet-110 + ICASC$_{A_{ch}}$. ........................................................... 33

3.1 An example of our coupled encoder-decoder framework result, simultaneously predicting the face regions (white bounding boxes) and the face fiducial points (white dots). The false positive responses for landmark localization are effectively suppressed by the coupled face detection task, in which regions marked by the red dash bounding boxes are classified as non-face. ........................................................... 36
3.2 The Illustration of coupled encoder and decoder network. (a) illustrates the encoder and decoder layers \((f_{ENC} \text{ and } f_{DENC})\), which consists of convolution(Conv), max pooling, up-sample pooling and fully convolutional layers. (b) shows the coupling structure in which we collect the feature maps from \(f_{ENC} \text{ and } f_{DENC}\) for procedure (c). (c) shows the coupled cascade face detection net \(f_{CLS}\), where the sliding window and ROI pooling are applied on the feature maps to generate the feature representations. The proposals, classified as positive (green bounding boxes) in all three stages, are collected and non-maximum suppressed as face regions.

3.3 The architecture of \(f_{REG}\). ROI pooling is applied on the foreground channel and the last encoder’s feature layer, yielding feature vectors \(f_l\) and \(f_g\). \(f_{REG}\) utilizes the concatenation of \(f_l\) and \(f_g\) to predict 2D coordinates for \(N\) landmarks.

3.4 The face landmark localization results on three benchmarks, Helen (left column), AFW (middle column) and LFPW (right column). The top row illustrates the localization errors for seven facial components, left corner of left eye (le_lc), right corner of left eye (le_rc), left corner of right eye (re_lc), right corner of right eye (re_rc), mouth left corner (m_lc), mouth right corner (m_rc) and nose tip. The bottom row shows the cumulative curves of relative mean errors, where the horizontal axis is the normalized distance with respect to the inter-ocular distance and the vertical axis is the proportion of images in the dataset.
3.5 The evaluation of face detection on datasets of FDDB (left) and AFW (right). On the FDDB dataset we compare our performance with the state-of-the-art methods including: CascadeCNN [6], Joint Cascade [90], DDFD [69], ACF-multiscale [107], PEP-Adapt [108], Boosted Exemplar [67], HeadHunter [57], Pico [109], Viola-Jones [58], TSM [60], HPM [89]. On the AFW dataset, comparison methods includes: HeadHunter [57], DPM [57], SquaresChnFtrs-5 [57], Shen et al. [66], TSM [60], and three commercial applications, Face++, face.com and Picasa.

3.6 The examples of faces in the dataset with low resolution, where face landmark localization is challenging.

3.7 Face examples of the multi-sensor database, which are collected using various cell phones and photographed in different distances.

3.8 Failure cases of finding the face area for FR under challenge conditions.

3.9 Qualitative results of face detection and landmark localization.

4.1 We leverage the trailer shots to estimate the attention scores of individual shots in the full-length movie, which indicate the “trailerness” of the shots and can be used as weak supervision to model the contrastive relation between the key and non-key moments in the feature space.

4.2 Duration distribution for the trailer and non-trailer shots indicates that the duration of the two kinds of shots is similar.

4.3 Left: overview of the proposed CCANet. We use the Co-Attention between the trailer and movie as the weak supervision and propose Contrastive Attention to augment the feature representations such that the trailer shots can stand out from the non-trailer shots in the feature space. Right: the details of Contrastive Attention module. ⨂ indicates matrix multiplication and “Concat” stands for vector concatenation.
4.4 **Left:** Rank@N. We calculate AP within every $N$ consecutive shots in a full-length movie and average them as the overall performance metric, offering a local-view on the ranking performance. The top row lists trailer (blue) and non-trailer (grey) shots in a movie along the timeline before ranking. The middle and bottom illustrate the ideal Rank@N results. **Right:** the “hard” annotation brings about ambiguity in the labels. A trailer shot and its four visually similar movie shots are shown. The movie shot marked by the green border is labeled as positive and the rest shots are negative. 66

4.5 Performance variance with respect to $\lambda$. We change the $\lambda$ value and report the performance of the proposed CCANet. The evaluation metric is Rank@10. 68

4.6 The normalized Co-Attention scores between Trailer and Movie shots. 72

4.7 The UMAP visualization of features over domains “Surfing” and “Gymnastics” (best view in color, zoom in). 72

5.1 The attention maps are generated via weighted combination of the feature maps at the specific layers. The weights measure the importance of the features, computed according to the gradients, where we take derivative of the class score w.r.t the feature maps. 77

5.2 The details of C3D architectures. 80

5.3 We compare what the model attends to known Action Units which are useful in communication research regarding face and head. The attended facial cues are coded as facial action units (AU). All examples are from spies. 85

5.4 What the model attends to for ”Spy” vs. ”Villager”. We show the comparison of attention maps between spy and villager. The model can attend to small facial movements related to deception, like eye blinking in the bottom left row (spies). And at the top row (spies and villagers), the model detects fake and real smiles to classify the two types of players’ roles correctly. 86

xvi
CHAPTER 1
INTRODUCTION

Tremendous interest in deep learning has emerged [1]. The most established algorithm among various deep learning models is the convolutional neural network (CNN), which has been a dominant approach in computer vision research. Specifically, CNN has demonstrated marginal performance boost over most of non deep learning approaches on several vision tasks [2, 3], including image classification [4], objection detection [5], face detection [6], image segmentation [7] and video understanding [8] etc.

Though CNN achieves astonishing results, there are still problems that need to be addressed. First, the deep CNN models are perceived as a “black-box” with a lack of understanding of the internal function. When we investigate the reason why the model fails, its lack of decomposability into intuitive and understandable components makes it hard to interpret [9].

Second, the success of deep learning relies on supervised training the models on the large-scale data, which requires humans to create massive annotations[10]. For example, the award-winning image classification neural network architecture, Residual Network [2] is trained on ImageNet dataset [4], which contains about 14 million images and 22K annotated image concepts. Additionally, annotators might be required to have expert domain knowledge. In VOC2012 [11] and COCO [12] dataset, the annotated segmentation masks make it possible to supervised learn an image segmentation model, and drawing the fine-grained segmentation masks results in very high annotation cost.

In this dissertation, we mainly address that attention mechanisms [13, 3, 14, 9] can play significant roles in dealing with the challenges mentioned above. Until recent years, the attention mechanism emerges as arguably one of the most powerful concepts in the deep learning field nowadays. It is developed based on an intuition that models “attend to” a
Figure 1.1: The class-specific attention highlights the important pixels for CNN recognizing specific categories. The middle indicates the input image. The leftmost depicts the attention for the “Person” category; and the rightmost is the attention for “Bird”.

certain part when processing images. Specifically, Figure 1.1 shows an example of class-specific attention, where the attention mechanisms generate the activation, which allows the network to focus on specific pixels of an input image. That activation highlights the important information for models to make the final decision, which motivates us to utilize attention as the critical method to propose algorithms tackling visual learning problems in three aspects. First, we take advantage of the attention mechanism to get a visual explanation of CNN, which can reason the model prediction. Second, the generated attention might be incorporated into the learning process, resulting in a more robust model trained with the guidance of attention maps and a framework conducting multi-tasks. Third, we leverage the “attended” parts as the weak supervision to generate positive/negative samples for training the model, resulting in reduced human annotation effort, and meanwhile, the attention is applied to augment the feature representation to maximize the contrast between positive and negative samples.

1.1 Visual Explanation

Some recent effort has been expended in visualizing internal representations of CNNs to interpret the model. In particular, the developments in gradient-based attention modeling have seen attention maps emerge as a powerful tool for interpreting convolutional neural networks. There are three approaches that made progress by proposing the class-specific
Figure 1.2: The Grad-CAM of different feature layers for two categories. The top indicates the attention maps for the category “common iguana” and bottom is “coho”. For each category, the attentions from higher to lower layers are listed left to right. For the same category, the model attention shift across the layers.

attention [14, 9, 15]. CAM [14] generated the class activation maps highlighting the task-relevant region by replacing fully-connected layers with convolution and global average pooling. A drawback of CAM is inflexibility, requiring retraining of classifiers and feature maps to directly precede softmax layers, hindering its applicability to any feature layers. Grad-CAM [9] was proposed to address these issues, where without retraining and changing the network architecture, class activation maps were generated by a weighted combination of feature maps in various channels. The weights were computed by averaging the gradient of the final prediction with respect to the pixels in the feature map. However, we observe that such simple averaging is unable to measure channel importance properly, resulting in substantial attention inconsistency among various feature layers. While Grad-CAM++ [15] proposed a better class activation map by modifying the way weights are computed, its high computational cost in calculating the second and third derivatives makes it impractical to be used during model training. Specifically, as shown in the leftmost attention maps of two rows in Figure 1.2, Grad-CAM generates the class-specific activation maps of the last feature layer, highlighting the essential pixels where the model attends to for the target objects, which indicate the spatial locations of the target objects. However, Grad-CAM [9] fails to produce intuitively satisfying attention maps from the inner layer, where the same drawbacks are observed in the
other two attention mechanisms. To illustrate this better, we depict the Grad-CAM of two categories in Figure 1.2. In this figure, the attention across different layers is not consistent, indicating that we need to propose a new mechanism to generate improved attention maps.

1.2 Incorporate Attention into Learning Process

Several recent methods [16, 17, 3, 18] have attempted to incorporate attention processing to improve the performance of CNNs in large-scale image classification. While these methods use attention for downstream classification, they do not explicitly use category-specific attention to train a model towards the image classification objective. The category-specific attention mechanisms produce attention maps specific to each class of interest, whereas, for image classification, we want these attention maps to be discriminative across categories. The model can tell various object categories apart. Our intuition is shown in Figure 1.3, where the separable attention maps of different categories can lead to better model interpretability for classification and improved classification performance. To this end, we devise novel objective functions to guide our model towards discriminative attention across different categories, leading to improved classification performance. In our work, to the best of our knowledge, we are the first to make category-specific attention end-to-end trainable with the specific goal of guiding the model with the supervisory signals from attention. Furthermore, we aim to have our method flexible, which can be considered as an add-on module to existing image classification architectures without needing any architectural change, unlike other methods [16, 17, 3, 18].

Additionally, we address that the crucial information in the feature spaces attended by the CNN model can be shared among similar vision tasks. For example, in face detection and landmark localization, the feature attended by the model for localizing the facial landmarks, e.g., eyes, nose, and mouth, is intuitively able to describe “what is a face”. Those features might be used to detect faces. It motivates us to use attention as the key to designing a single model for joint learning multiple vision tasks.
Figure 1.3: The model with discriminative attention can tell various object categories apart. Given an input image with ground truth label “boom box”, we show the category-specific attention maps for top-4 predictions: “headphone”, “video projector”, “boom box”, and “toaster”. The attention map and prediction score for ground truth class “boom box” are marked by red bounding box and bar respectively. The less discriminative attention from vanilla CNN highlights very similar pixels, confusing the model to recognize the image as “boom box”. The more separable attention provides the better model interpretability, where the model attends different crucial pixels across various categories, eliminating the confusion and leading to better model discriminability.

1.3 Reduce Annotation Budget

With the popularity of crowd-sourcing platforms such as Amazon Mechanical Turk, most large-scale computer vision datasets are annotated via public crowd-sourcing platforms [10]. Most natural image annotation tasks, e.g., image classification, are simple visual perception tasks, and therefore the annotation cost is reduced via a crowd-sourcing platform. However, some tasks still involve high annotations cost where fine-grained labels are required. For example, regards object detection and image segmentation, annotators are asked to precisely provide a tight bounding box to localize objects and draw contours along the objects to get image segmentation masks. We can utilize attention mechanisms to conduct weakly-supervised object localization and semantic segmentation with only image-level annotation [19, 7, 20, 5], reducing the burden of annotations in the level of bounding-boxes.
and pixels. The main goal is to have the model’s attention covering the whole target object, requiring the attention mechanism to be discriminative to identify the most semantic information.

Additionally, since visual perception is a universal capability, these annotation tasks can be reliably completed by normal annotators without any specialized domain knowledge [10]. However, for some specific applications, e.g., movie trailer moment detection or facial behaviors analysis for deception, annotation requires experts with domain knowledge. With regard to movie trailer moment detection, annotating trailer moments is very challenging as the selection of trailer moments might attribute to various factors such as emotion, environment, story-line, or visual effects, which requires the annotators to watch the movies and trailers, and provide precise timestamps of movies segments selected as trailer moments. To analyze the facial behaviors for deception, it needs social communication experts watch the videos and identify those micro facial actions which tell whether deception occurs or not. We propose using the attention mechanism as the supervision signal to resolve this issue by learning to discover the positive and negative samples automatically. Meanwhile, inspired by class-discriminative attention, we proposed learning the feature representations under attention guidance, where the comparative contrast between features of the positive and negative samples is maximized.

1.4 Contribution of the Dissertation

We mainly focused on addressing how to apply attention mechanism to the learning process, which deals with the challenges mentioned above. Our contributions are summarized below. First, we address the drawbacks of the existing gradient-based class-specific activation maps and provide an attention mechanism that has better localizability and avoids higher-order derivative computation. Our proposed attention mechanism’s effectiveness is examined in the scenario of weakly-supervised image segmentation on the public benchmark, indicating superior performance over the previous approaches.
Second, we address incorporating the attention into the learning process from two aspects: 1) we propose a novel framework that integrates class-specific attention in the conventional learning process, providing the learning objectives for training CNN, which results in the discriminative attention and the improved classification performance against the learning baselines without attention guidance on several benchmarks; 2) we unitize the attention as the critical method to bridge different visual tasks into a unified learning process, where the proposed approach achieves joint face detection and landmark localization, demonstrating improved performance under face-in-the-wild conditions.

Third, we address the problem of using attention as weak supervision to generate positive and negative samples. And we apply attention to augment the feature representations such that the comparative contrast between features of the positive and negative samples is maximized. The proposed algorithm is applied to the automatic detection of trailer moments from full-length movies without the need for human annotation. Our approach shows superior performance over the state-of-the-art. Additionally, to our best knowledge, we are the first to collect a trailer moment detection dataset to facilitate this research direction.

Last, we applied our attention mechanisms to analyze facial videos collected from a version of the board game *The Resistance*. In this game, players from various countries were randomly and secretly assigned to play deceivers (called "Spies"), or truth-tellers (called "Villagers"). Our approach aims to recognize “who is a spy” v.s."who is a villager" and discover which frames and facial expressions (AUs) contributed most to CNN’s class decision. Our approach demonstrates that we are on par with human recognition of spies vs. villagers. Also, our attention mechanism can attend and discover the frames and associated facial action units (AUs) consistent with the current communication theory about deception.

1.5 Outline of the Dissertation

The content of this thesis is arranged into the following chapters.

In Chapter 2, we propose a channel-weighted attention, which has better localizability
and avoids higher-order derivatives computation, compared to existing attention approaches and addresses the problem of Learning with Attention Separability and Consistency. Our attention-guided training is flexible to be added to any network without changing net architectures, which is an end-to-end procedure, reducing visual confusion. The conducted experiments show performance-boosting on various image-classification tasks, including classifying images in the setting of medium-scale, large-scale, fine-grained, and multi-class.

In Chapter 3, we show the possibility that the feature space’s attention plays a significant role in bridging the multiple vision tasks into a unified learning framework. Specifically, we proposed a coupled encoder-decoder neural network to detect faces and localize landmarks jointly.

In Chapter 4, we utilize Co-Attention as supervision to learn a rank model, which does not require expensive human-annotations. Practically, the proposed Co-Attention is used to address the problem of learning the trailer moments from the full-length movie. Additionally, we introduce the Contrastive Attention to augment the video features, equipping the model with the capacity of capturing the contrastive relation between the trailer and non-trailer moments.

Chapter 5 presents an attention-based neural network that discovers through learning in a video sequence the most discriminative frames and related pixel probabilities. We applied our method to facial videos of a variant of the Resistance game collected in various countries. We demonstrated for the first time that it is possible to discover the frames and Action Units (AUs) that contributed the most to the neural network’s decision on several hours of video testing.

Chapter 6 draws the conclusion and discuss the current limitation and future directions.
CHAPTER 2
LEARNING WITH ATTENTION SEPARABILITY AND CONSISTENCY

2.1 Introduction

Visual recognition has seen tremendous progress in the last few years, driven by recent advances in convolutional neural networks (CNNs) [2, 3]. Understanding their predictions can help interpret models and provide cues to design improved algorithms.

Recently, class-specific attention has emerged as a powerful tool for interpreting CNNs [14, 9, 15]. The big-picture intuition that drives these techniques is to answer the following question- *where is the target object in the image?* Some recent extensions [7] make attention end-to-end trainable, producing attention maps with better localizability. While these methods consider the localization problem, this is insufficient for image classification, where the model needs to be able to tell various object classes apart. Specifically, existing methods produce attention maps corresponding to an individual class of interest that may not be *discriminative* across classes. Our intuition, shown in Figure 2.1, is that such separable attention maps can lead to improved classification performance. Furthermore, we contend that false classifications stem from patterns across classes which confuse the model, and that eliminating these confusions can lead to better model discriminability. To illustrate this, consider Figure 2.2 (a), where we use the VGG-19 model [21] to perform classification on the ILSVRC2012 [4] dataset, we collect failure cases and generate the attention maps via Grad-CAM [9] and we show the top-5 predictions. Figure 2.2 (a) depicts that, while the attention maps of the last feature layer are reasonably well localized, there are large overlapping regions between the attention of the ground-truth class (marked by red bounding boxes) and the false positives, demonstrating the problem, and the need for *discriminative attention*. 
To overcome the above attention-map limitations, we need to address two key questions: (a) \textit{can we reduce visual confusion, i.e., make class-specific attention maps separable and discriminative across different classes?}, and (b) \textit{can we incorporate attention discriminability in the learning process in an end-to-end fashion?} We answer these questions in a principled manner, proposing the first framework that makes attention maps class discriminative. Furthermore, we propose a new attention mechanism to guide model training towards attention discriminability, which provides end-to-end supervisory signals by explicitly enforcing attention maps of various classes to be separable.

Attention separability and localizability are key aspects of our proposed learning framework for image classification. Non-separable attention maps from the last layer, as shown in Figure 2.2 (a), prompted us to look “further inside” the CNN and Figure 2.2 (b) shows atten-
Figure 2.2: Grad-CAM [9] attention maps of the VGG-19 [21] top-5 predictions. Predictions with red-bounding boxes correspond to the ground-truth class. (a) Ground-truth class attention maps from the last layer (Conv5) have a large overlap with false positives (top-1 predictions). (b) Inner-layer attention maps (Conv4) are more separable than their last-layer counterparts.

tion maps from an intermediate layer. This illustration shows that these inner-layer attention maps are more separable than those from the last layer. However, the inner-layer attention maps are not as well-localized as the last layer. So, another question we ask is—can we get the separability of the inner-layer attention and the localization of the last-layer attention at the same time? Solving this problem would result in a “best-of-both-worlds” attention map that is separable and localized, which is our goal. To this end, we also propose an explicit mechanism that enforces the ground-truth class attention to be cross-layer consistent.

We conduct experiments on five competitive benchmarks (CIFAR-100 [22], Caltech-256 [23], ILSVRC2012 [4], CUB-200-2011 [24] and PASCAL VOC 2012 [11]), showing performance improvements of 3.33%, 1.64%, 0.92%, 4.8%, and 5.73%, respectively.
2.2 Related Work

**Visualizing CNNs.** Much recent effort has been expended in visualizing internal representations of CNNs to interpret the model. Erhan *et al.* [25] synthesized images to maximally activate a network unit. Mahendran *et al.* [26] and Dosovitskiy *et al.* [27] analyzed the visual coding to invert latent representations, performing image reconstruction by feature inversion with an up-convolutional neural network. In [28, 29, 30], the gradient of the prediction was computed w.r.t. the specific CNN unit to highlight important pixels. These approaches are compared in [31, 9]. The visualizations are fine-grained but not class-specific, where visualizations for different classes are nearly identical [9].

Our framework is inspired by recent works [14, 9, 15] addressing class-specific attention. CAM [14] generated class activation maps highlighting task-relevant regions by replacing fully-connected layers with convolution and global average pooling. Grad-CAM [9] solved CAM’s inflexibility where without changing the model architecture and retraining the parameters, class-wise attention maps were generated by means of gradients of the final prediction w.r.t. pixels in feature maps. However, we observe that directly averaging gradients in Grad-CAM [9] results in the improper measurement of channel importance, producing substantial attention inconsistency among various feature layers. Grad-CAM++ [15] proposed to introduce higher-order derivatives to capture pixel importance, while its high computational cost in calculating the second- and third-order derivatives makes it impractical to be used during training.

CBAM [18, 33] modified the SE module to exploit both spatial and channel-wise attention. Jetley et al. [3] estimated attentions by considering the feature maps at various layers in the CNN, producing a 2D matrix of scores for each map. The ensemble of output scores was then used for class prediction. While these methods use attention for downstream classification, they do not explicitly use class-specific attention as part of model training for image classification.

Our work, to the best of our knowledge, is the first to use class-specific attention to produce supervisory signals for end-to-end model training with attention separability and cross-layer consistency. Furthermore, our proposed method can be considered as an add-on module to existing image classification architectures without needing any architectural change, unlike other methods [16, 17, 3, 18]. While class-specific attention has been used in the past for weakly-supervised object localization and semantic segmentation tasks [19, 7, 5, 20], we model attention differently. The goal of these methods is singular - to make the attention well localize the ground-truth class, while our goal is two-fold - good attention localizability as well as discriminability. To this end, we devise novel objective functions to guide model training towards discriminative attention across different classes, leading to improved classification performance as we show in the experiments section.

2.3 Approach

In Figure 2.3, we propose “Improving Classification with Attention Separation and Consistency” (ICASC), the first end-to-end learning framework to improve model discriminability for image classification via attention-driven learning. The main idea is to produce separable attention across various classes, providing supervisory signals for the learning process. The motivation comes from our observations from Figure 2.2 that the last layer attention maps computed by the existing methods such as Grad-CAM [9] are not class-separable, although they are reasonably well localized. To address this problem, we propose the attention separation loss $L_{AS}$, a new learning objective to enforce attention discriminability.
Additionally, we observe from Figure 2.2 that inner layer attention at higher resolution has the potential to be separable, which suggests we consider both intermediate and the last layer attention to achieve separability and localizability at the same time. To this end, we propose the attention consistency loss $L_{AC}$, a new cross-layer attention consistency learning objective to enforce consistency among inner and last layer attention maps. Both proposed learning objectives require that we obtain reasonable attention maps from the inner layer. However, Grad-CAM [9] fails to produce intuitively satisfying inner layer attention maps. To illustrate this, we depict two Grad-CAM [9] examples in Figure 2.4, where we see the need for better inner layer attention. To this end, we propose a new channel-weighted attention mechanism $A_{ch}$ to generate improved attention maps (explained in Sec. Subsection 2.3.1). We then discuss how we use them to produce supervisory signals for enforcing attention separability and cross-layer consistency.
Figure 2.4: The Grad-CAM [9] attentions of different VGG-19 [21] feature layers for the 'tench' class. In both rows, the target is the fish while the model attention shifts across the layers.

2.3.1 Channel-weighted Attention $A_{ch}$

Commonly-used techniques to compute gradient-based attention maps given class labels include CAM [14], Grad-CAM [9], and Grad-CAM++ [15]. We do not use CAM because (a) it is inflexible, requiring network architecture modification and model re-training, and (b) it works only for the last feature layer.

Compared to CAM [14], Grad-CAM [9] and Grad-CAM++ [15] are both flexible in the sense that they only need to compute the gradient of the class prediction score w.r.t. the feature maps to measure pixel importance. Specifically, given the class score $Y^c$ for the class $c$ and the feature map $F^k$ in the $k$-th channel, the class-specific gradient is determined by computing the partial derivative $(\partial Y^c)/(\partial F^k)$. The attention map is then generated as $A = ReLU(\sum_k \alpha_k^c F^k)$, where $\alpha_k^c$ indicates the importance of $F^k$ in the $k$-th channel. In Grad-CAM [9], the weight $\alpha_k^c$ is a global average of the pixel importance in $(\partial Y^c)/(\partial F^k)$:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial F^k_{ij}}$$

(2.1)

where $Z$ is the number of pixels in $F^k$. Grad-CAM++ [15] further introduces higher-order derivatives to compute $\alpha_k^c$ so as to model pixel importance.
Although Grad-CAM [9] and Grad-CAM++ [15] are more flexible than CAM [14], they have several drawbacks that hinder their use as is for our purposes of providing separable and consistent attention guidance for image classification. First, there are large attention shifts among attention maps of different feature layers in Grad-CAM [9] which are caused by negative gradients while computing channel-wise importance. A key aspect of our proposed framework ICASC is to exploit the separability we observe in inner layer attention in addition to good localization from the last layer attention. While we observe relatively less attention shift with Grad-CAM++ [15], the high computational cost of computing higher-order derivatives precludes its use in ICASC since we use attention maps from multiple layers to guide model training in every iteration.

To address these issues, we propose channel-weighted attention $A_{ch}$, highlighting the pixels where the gradients are positive. In our exploratory experiments, we observed that the cross-layer inconsistency of Grad-CAM [9], noted above, is due to negative gradients from background pixels. In Grad-CAM [9], all pixels of the gradient map contribute equally to the channel weight (Eq. Equation 2.1). Therefore, in cases where background gradients dominate, the model tends to attend only to small regions of target objects, ignoring regions that are important for class discrimination.

We are motivated by prior work [15, 30, 29] that observes that positive gradients w.r.t. each pixel in the feature map $F^k$ strongly correlate with the importance for a certain class. A positive gradient at a specific location implies increasing the pixel intensity in $F^k$ will have a positive impact on the prediction score, $Y^c$. To this end, driven by positive gradients, we propose a new channel-weighted attention mechanism $A_{ch}$:

$$A_{ch} = \frac{1}{Z} ReLU(\sum_k \sum_i \sum_j ReLU(\frac{\partial Y^c}{\partial F^k_{ij}})F^k)$$

(2.2)

Our attention does not need to compute higher-order derivatives as in Grad-CAM++ [15], while also resulting in well-localized attention maps with relatively less shift unlike Grad-CAM [9], as shown in Figure 2.5.
Figure 2.5: The comparison of attention maps from different VGG-19 [21] layers. Ours has less attention shift than Grad-CAM [9]. In the marked areas, ours attends to the target objects, i.e. bird, while Grad-CAM [9] tends to highlight the background pixels.

2.3.2 Attention Separation Loss $L_{AS}$

We use the notion of attention separability as a principled part of our learning process and propose a new learning objective $L_{AS}$. Essentially, given the attention map of a ground-truth class $A^T$ and the most confusing class $A^{Conf}$, where $A^{Conf}$ comes from the non-ground truth class with the highest classification probability, we enforce the two attentions to be separable. We reflect this during training by quantifying overlapping regions between $A^T$ and $A^{Conf}$, and minimizing it. To this end, we propose $L_{AS}$ which is defined as:

$$L_{AS} = 2 \cdot \frac{\sum_{ij} (\min(A^T_{ij}, A^{Conf}_{ij}) \cdot Mask_{ij})}{\sum_{ij} (A^T_{ij} + A^{Conf}_{ij})},$$

(2.3)

where the $\cdot$ operator indicates scalar product, and $A^T_{ij}$ and $A^{Conf}_{ij}$ represent the $(i, j)^{th}$ pixel in attention maps $A^T$ and $A^{Conf}$ respectively. The proposed $L_{AS}$ is differentiable which can be used for model training.

Additionally, to reduce noise from background pixels, we apply a mask to focus on pixels within the target object region for the $L_{AS}$ computation. In Eq. Equation 2.3, $Mask$ indicates the target object region generated by thresholding the attention map $A^T$ from the last layer:

$$Mask_{ij} = \frac{1}{1 + exp(-\omega(A^T_{ij} - \sigma))},$$

(2.4)
where we empirically choose values of $\sigma$ and $\omega$ to be $0.55 \times \max (A_{ij}^T)$ and 100 respectively.

The intuition of $L_{AS}$ is illustrated in Figure 2.6. If the model attends to the same or overlapped regions for different classes, it results in visual confusion. We penalize the confusion by explicitly reducing the overlap between the attention maps of the target and the most confusing class. Specifically, we minimize $L_{AS}$, which is differentiable with values ranging from 0 to 1.

The proposed $L_{AS}$ is an add-on module for training a model without changing the network architecture. Besides applying $L_{AS}$ to the last feature layer, we can also compute $L_{AS}$ for other layers, which makes it possible to analyze model attention at various scales.

While the proposed $L_{AS}$ helps enforce attention separability, it is not sufficient for image classification since inner layer attention maps are not as spatially well-localized as the last layer. We set out to achieve an attention map to be well-localized and class-discriminative, and to this end, we propose a cross-layer attention consistency objective $L_{AC}$ that enforces the target attention of inner layers to be similar to that of the last layer.
2.3.3 Attention Consistency Loss $L_{AC}$

In higher layers (layers closer to output), the model attention captures more semantic information, covering most of the target object \cite{14, 9, 15}. For the intermediate layers with the smaller receptive fields of the convolution kernels, the model attends to more fine-grained patterns as shown in Figures Figure 2.4 and Figure 2.5. Compared to higher-layer attention, lower-layer attention contains more noise, highlighting background pixels.

To address the issues, we propose the attention consistency loss $L_{AC}$ to correct the attention so that the highlighted fine-grained attention is primarily localized in the target region:

$$L_{AC} = \theta - \frac{\sum_{ij} (A_{ij}^{in} \cdot Mask_{ij})}{\sum_{ij} A_{ij}^{in}},$$  \hspace{1cm} (2.5)

where $A_{ij}^{in}$ indicates attention maps from the inner feature layers, $Mask_{ij}$ (defined in Eq. Equation 2.4) represents the target region, and $\theta$ is set to 0.8 empirically. As can be noted from Eq. Equation 2.5, the intuition of $L_{AC}$ is that by exploiting last layer attention’s good localizability, we can guide the inner layer attention to be chiefly concentrated within the target region as well. This guidance $L_{AC}$ helps maintain cross-layer attention consistency.

An another interpretation of our proposed $L_{AC}$ can be provided by the analogy to the auxiliary classifiers proposed by C. Szegedy et al. in \cite{34}. While both $L_{AC}$ and auxiliary classifiers provide additional regularization to inner layers, the focus of auxiliary classifiers is to intensify the gradient to prevent vanishing gradients, and the motivation of $L_{AC}$ is to ensure high response activation from inner layers are localized within the target region and not elsewhere, e.g., background.

2.3.4 Overall Framework ICASC

We apply the constraints of attention separability and cross-layer consistency jointly as supervisory signals to guide end-to-end model training, as shown in Figure 2.3. Firstly, we
compute inner-layer attention for the loss $L_{AS}^{in}$ with the purpose of enforcing inner-layer attention separability. For example, with ResNet, we use the last convolutional layer in the penultimate block. We empirically adopt this to compute $L_{AS}^{in}$ in consideration of the low-level patterns and semantic information addressed by the inner-layer attention. In Figure 2.5, this inner-layer attention, with twice resolution as the last layer, highlights more fine-grained patterns while still preserving the semantic information, thus localizing the target object. We also apply the $L_{AS}$ constraint on the attention map from the last layer, giving us $L_{AS}^{la}$. Secondly, we apply the cross-layer consistency constraint $L_{AC}$ between the attention maps from these two layers. Finally, for the classification loss $L_C$, we use cross-entropy and multilabel-soft-margin loss for single and multi-label image classification respectively. The overall training objective of ICASC, $L$, is:

$$L = L_C + L_{AS}^{in} + L_{AS}^{la} + L_{AC}$$ (2.6)

ICASC can be used with available attention mechanisms including Grad-CAM [9] and $A_{ch}$. We use ICASC$_{Grad-CAM}$ and ICASC$_{A_{ch}}$ to refer to our framework used with Grad-CAM [9] and $A_{ch}$ as the attention mechanisms respectively.

### 2.4 Experiments

Our experiments contain two parts, (a) evaluating the class discrimination ability of various attention mechanisms, and (b) demonstrating the effectiveness of the proposed ICASC by comparing it with the corresponding baseline model (having the same architecture) without the attention supervision.

#### 2.4.1 Evaluating Class Discriminability

We first evaluate class-discriminability of our proposed attention mechanism $A_{ch}$ by measuring both localizability (identifying target objects) and discriminability (separating different
Figure 2.7 shows that $A_{ch}$ (ours) has better localization for the two classes, “Bird” and “Person” compared to Grad-CAM and Grad-CAM++. In “Bird,” both Grad-CAM and Grad-CAM++ highlight false positive pixels in the bottom-left area, whereas in “Person,” Grad-CAM++ attends to a much larger region than Grad-CAM and $A_{ch}$. Figure 2.8 qualitatively demonstrates better class-discriminative segmentation maps using $A_{ch}$. In Figure 2.8 top row, as expected for a single object, all methods, including $A_{ch}$, show good performance localizing the sheep. The second row shows that Grad-CAM covers more noise pixels of
Figure 2.8: Segmentation masks generated from attention maps by DeepLab [35] (best view in color, zoom in). From left to right: the Input Image, Ground Truth, Grad-CAM, Grad-CAM++ and ours.

<table>
<thead>
<tr>
<th>Attention Mechanism</th>
<th>Segmentation Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grad-CAM [9]</td>
<td>56.65</td>
</tr>
<tr>
<td>Grad-CAM++ [15]</td>
<td>51.70</td>
</tr>
<tr>
<td>$A_{ch}$ (ours)</td>
<td>57.97</td>
</tr>
</tbody>
</table>

Table 2.1: Results on Pascal VOC 2012 segmentation validation set.

the grassland, while $A_{ch}$ produces similar results as Grad-CAM++, both of which are better than Grad-CAM in identifying multiple instances of the same class. Finally, for multi-class images in the last row, $A_{ch}$ demonstrates superior results when compared to both Grad-CAM and Grad-CAM++. Specifically, $A_{ch}$ is able to tell the motorcycle, the person, and the car apart in the last row. We also obtain the quantitative results and report the score from the Pascal VOC Evaluation server in Table 2.1, where $A_{ch}$ outperforms both Grad-CAM and Grad-CAM++. The qualitative and quantitative results show that $A_{ch}$ localizes and separates target objects better than the baselines, motivating us to use $A_{ch}$ in ICASC, which we evaluate in section Subsection 2.4.2.
2.4.2 Evaluating ICASC for Image Classification

Implementation Details

We conduct image classification experiments on various datasets, consisting of three parts: generic image classification on CIFAR-100 ($D_{CI}$) [22], Caltech-256 ($D_{Ca}$) [23] and ILSVRC-2012 ($D_I$) [4], fine-grained image classification on CUB-200-2011 ($D_{CU}$) [24], and finally, multi-label image classification on PASCAL VOC 2012 ($D_P$) [11]. For simplicity, we use the shorthand in the parenthesis after the dataset names above to refer to each dataset and its associated task, and summarize all experimental parameters used in Table 2.2.

We perform all experiments using PyTorch [38] and NVIDIA Titan X GPUs. We use the same training parameters as those in the baselines proposed by the authors of the corresponding papers for fair comparison.

**CIFAR-100:** The image is padded by 4 pixels on each side, filled with 0 value resulting in a 40×40 image. A 32×32 crop is randomly sampled from an image or its horizontal flip, with the per-pixel RGB mean value subtracted. We adopt the same weight initialization method following [2] and train the ResNet using Stochastic Gradient Descent (SGD) [40] with a mini-batch size of 128. We use a weight decay of 0.0005 with a momentum of 0.9.

---

<table>
<thead>
<tr>
<th>Task</th>
<th>$D_{CI}$</th>
<th>$D_{Ca}$</th>
<th>$D_I$</th>
<th>$D_{CU}$</th>
<th>$D_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNA</td>
<td>RN-18</td>
<td>VGG</td>
<td>RN-18</td>
<td>RN-50</td>
<td>RN-18</td>
</tr>
<tr>
<td>WD</td>
<td>5e$^{-4}$</td>
<td>1e$^{-3}$</td>
<td>1e$^{-4}$</td>
<td>5e$^{-4}$</td>
<td>1e$^{-3}$</td>
</tr>
<tr>
<td>MOM</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>LR</td>
<td>1e$^{-1}$</td>
<td>1e$^{-2}$</td>
<td>1e$^{-1}$</td>
<td>1e$^{-3}$</td>
<td>1e$^{-2}$</td>
</tr>
<tr>
<td>BS</td>
<td>128</td>
<td>16</td>
<td>256</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>OPM</td>
<td>SGD</td>
<td>CCA</td>
<td>SGD</td>
<td>SGD</td>
<td>CCA</td>
</tr>
<tr>
<td># epoch</td>
<td>160</td>
<td>20</td>
<td>90</td>
<td>90</td>
<td>20</td>
</tr>
</tbody>
</table>

*Table 2.2: Experimental (exp.) settings used in this paper. VGG, RN-18, RN-50, and RN-101 denote VGG-19 [28], ResNet-18 [2], ResNet-50, and ResNet-101, respectively. We use the same parameters as the references in the last row unless otherwise specified, putting more details in the supplementary material. Acronyms: BNA: base network architecture; WD: weight decay; MOM: momentum; LR: initial learning rate; BS: batch size; OPM: optimizer; SGD: stochastic gradient descent [40]; CCA: cyclic cosine annealing [41].*
and set the initial learning rate to 0.1. The learning rate is divided by 10 at 81 and 122 epochs. The training is terminated after 160 epochs.

**Caltech-256**: There is no official training/testing data split. We follow the work in [23] to randomly select 25 images per category as the testing set and 30, 60 images per category as training. We remove the last (257-th) category “clutter,” keeping the 256 categories which describe specific objects. We use VGG-19 [28] and ResNet-18 [2] as the baseline models. For the training of both the baseline and our proposed method, we use a weight decay of 0.001 with a momentum of 0.9 and set the initial learning rate to 0.01. To speed up the model training, we adopt cyclic cosine annealing [41] with a cycle of one to train the network for 20 epochs.

**ILSVRC2012**: We conduct large-scale image classification experiments using the ImageNet ILSVRC2012 dataset [4]. The evaluation is conducted on the images of the ILSVRC-2012 validation set. We use ResNet-18 [2] as the baseline model. We use SGD [40] with a mini-batch size of 256 to train the network. The initial learning rate is set as 0.1 and weight decay of 0.0001 with a momentum of 0.9. The learning rate is divided by 10 at 30 and 60 epochs. The training is terminated after 90 epochs.

**CUB-200-2011**: We follow the training pipeline from [39] to choose ResNet-50 and ResNet-101 as the baseline models. The input images are resized to $448 \times 448$ for both training and testing and we apply standard augmentation for training data, *i.e.* mirror, and random cropping. The SGD [40] is used to optimize the networks. The learning rate is decayed by 0.1 after 30 and 60 epochs.

**PASCAL VOC 2012**: We evaluate the Multi-class image classification performance on PASCAL VOC 2012 dataset. We use ResNet-18 with the Multi-Label-Soft-Margin loss as our baseline model. Cyclic cosine annealing [41] with the cycle of 1 is used to speed up the training. The total number of training epochs is 20.
Figure 2.9: The KS-Chart on the CUB-200-2011 testing set. “Ours” stands for ResNet-50 + $L_{AS}^{in} + L_{AS}^{lo} + L_{AC}$ in Table 2.3.

Table 2.3: Ablation study on CUB-200-2011 ($\Delta=$performance improvement; “Top-1”: top-1 accuracy (%)).

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
<th>$\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>81.70</td>
<td>-</td>
</tr>
<tr>
<td>+ $L_{AS}^{in}$</td>
<td>85.15</td>
<td>3.45</td>
</tr>
<tr>
<td>+ $L_{AS}^{in} + L_{AC}$</td>
<td>85.77</td>
<td>4.07</td>
</tr>
<tr>
<td>+ $L_{AS}^{in} + L_{AS}^{lo} + L_{AC}$</td>
<td><strong>86.20</strong></td>
<td>4.50</td>
</tr>
</tbody>
</table>

Ablation Study

Table 2.3 shows an ablation study with the CUB-200-2011 dataset, which provides a challenging testing set given its fine-grained nature. We use the last convolutional layer in the penultimate block of ResNet-50 for computing $L_{AS}^{in}$ and the last layer attention map for $L_{AS}^{lo}$. We see that $L_{AS}^{in} + L_{AS}^{lo} + L_{AC}$ achieves the best performance. The results show that the attention maps from the two different layers are complementary: last-layer attention has more semantic information, well localizing the target object, and inner layer attention with higher resolution provides fine-grained details. Though the inner-layer attention is more
Figure 2.10: Qualitative results with CIFAR-100. We show top-5 predictions with classification scores given by ResNet-110 (top row) and ResNet-110 + ICASC_Ach (bottom row).

likely to be noisy than the last layer, $L_{AC}$ provides the constraint to guide the inner-layer attention to be consistent with that of the last layer and be concentrated within the target region.

We quantitatively measure the degree of visual confusion reduction with our proposed learning framework. Specifically, as shown in Figure 2.9, we compute Kolmogorov-Smirnov (KS) statistics [42] on the CUB-200-2011 testing set, measuring the degree of separation between the ground-truth (Target) class and the most confusing (Confused) class distributions [43]. We rank non-ground truth classes in descending order according to their classification probabilities and determine the most confusing class as the one ranked highest. In Figure 2.9, for the baseline model, the largest margin is 0.64 at the classification probability 0.51 whereas our proposed model has a KS margin of 0.74 at the classification probability 0.55. This demonstrates that our model is able to recognize 10% more testing samples with higher confidence when compared to the baseline.

**Generic Image Classification**

Tables Table 2.4-Table 2.6 (in all tables, △ indicates performance improvement of our method over baseline) show that the models trained with our proposed supervisory principles
Method | Top-1 | △
--- | --- | ---
ResNet-110 [44] | 72.78 | -
ResNet-110 with Stochastic Depth [44] | 75.42 | -
ResNet-164 (pre-activation) [44] | 75.63 | -
ResNet-110 + ICASC \(_{Grad-CAM}\) | 74.02 | 1.24
ResNet-110 + ICASC \(_{Ach}\) | **76.11** | **3.33**

Table 2.4: Image classification results on CIFAR-100.

<table>
<thead>
<tr>
<th>Method</th>
<th>N=30</th>
<th>N=60</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
</tr>
</tbody>
</table>
| RN-18 [2] | 76.77 | 92.48 | - | 80.01 | 94.12 | -
| RN-18 + ICASC\(_{Ach}\) | **78.01** | **92.87** | **1.24** | **81.32** | **94.57** | **1.31**
| VGG-19 [21] | 74.52 | 90.05 | - | 78.16 | 92.17 | -
| VGG-19 + ICASC\(_{Ach}\) | **75.60** | **90.85** | **1.08** | **79.80** | **93.25** | **1.64**

Table 2.5: Results on Caltech-256. “Top-5”: top-5 accuracy (%). “RN-18”: ResNet-18. “N”: # of training images per class. We follow [23] to randomly select 30 or 60 training images per class.

outperform the corresponding baseline models with a notable margin. The most noticeable performance improvements are observed with the CIFAR-100 dataset in Table 2.4, which shows that, without changing the network architecture, the top-1 accuracy of ResNet-110 with our proposed supervision outperforms the baseline model by 3.33%. Our supervised ResNet-110 also outperforms the one with stochastic depth and even the much deeper model with 164 layers. As can be observed from the qualitative results in Figure 2.10, ICASC\(_{Ach}\) equips the model with discriminative attention where the ground-truth class attention is separable from the confusing class, resulting in improved prediction.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
<th>Top-5</th>
<th>△</th>
</tr>
</thead>
</table>
| ResNet-18 [2] | 69.51 | 88.91 | -
| ResNet-18 + ICASC\(_{Ach}\) | **69.90** | **89.71** | **0.39**
| ResNet-18 + tenCrop [2] | 72.12 | 90.58 | -
| ResNet-18 + tenCrop + ICASC\(_{Ach}\) | **73.04** | **90.65** | **0.92**

Table 2.6: Results on ILSVRC2012.
Table 2.7: Results on CUB-200-2011. “No Extra Anno.” means not using extra annotation (bounding box or part) in training. “1-Stage” means the training is done in one stage.

<table>
<thead>
<tr>
<th>Method</th>
<th>No Extra Anno.</th>
<th>1-Stage</th>
<th>Top-1</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 [39]</td>
<td>✔</td>
<td>✔</td>
<td>81.7</td>
<td>-</td>
</tr>
<tr>
<td>ResNet-101 [39]</td>
<td>✔</td>
<td>✔</td>
<td>82.5</td>
<td>0.8</td>
</tr>
<tr>
<td>MG-CNN [47]</td>
<td>×</td>
<td>×</td>
<td>83.0</td>
<td>1.3</td>
</tr>
<tr>
<td>SPDA-CNN [45]</td>
<td>×</td>
<td>✔</td>
<td>85.1</td>
<td>3.4</td>
</tr>
<tr>
<td>RACNN [48]</td>
<td>✔</td>
<td>✔</td>
<td>85.3</td>
<td>3.6</td>
</tr>
<tr>
<td>PN-CNN [49]</td>
<td>×</td>
<td>×</td>
<td>85.4</td>
<td>3.7</td>
</tr>
<tr>
<td>RAM [50]</td>
<td>✔</td>
<td>×</td>
<td>86.0</td>
<td>4.3</td>
</tr>
<tr>
<td>MACNN + 2parts [46]</td>
<td>✔</td>
<td>✔</td>
<td>85.4</td>
<td>3.7</td>
</tr>
<tr>
<td>ResNet-50 + MAMC [39]</td>
<td>✔</td>
<td>✔</td>
<td>86.2</td>
<td>4.5</td>
</tr>
<tr>
<td>ResNet-101 + MAMC [39]</td>
<td>✔</td>
<td>✔</td>
<td><strong>86.5</strong></td>
<td>4.8</td>
</tr>
<tr>
<td>ResNet-50 + ICASC_{A_{ch}}</td>
<td>✔</td>
<td>✔</td>
<td>86.2</td>
<td>4.5</td>
</tr>
<tr>
<td>ResNet-101 + ICASC_{A_{ch}}</td>
<td>✔</td>
<td>✔</td>
<td><strong>86.5</strong></td>
<td>4.8</td>
</tr>
</tbody>
</table>

**Fine-grained Image Recognition**

For fine-grained image recognition, we evaluate our approach on the CUB-200-2011 [24], which contains 11788 images (5994/5794 for training/testing) of 200 bird species. We show the results in Table 2.7. We observe that training with our learning mechanism boosts the accuracy of the baseline ResNet-50 and ResNet-101 by 4.8% and 4.0% respectively. Our method achieves the best overall performance against the state-of-the-art. Furthermore, with ResNet-50, our method outperforms even the method that uses extra annotations (PN-CNN) by 0.8%.

ICASC_{A_{ch}} has better flexibility compared to the other methods in Table 2.7. The existing methods are specifically designed for fine-grained image recognition where, according to prior knowledge of the fine-grained species, the base network architectures are modified to extract features of different objects parts [45, 46, 39]. In contrast, ICASC_{A_{ch}} needs no prior knowledge and works for generic image classification without changing the network architectures.
Table 2.8: Image classification results on Pascal VOC 2012.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC Score</th>
<th>AP (%)</th>
<th>△</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-18 [2]</td>
<td>0.976</td>
<td>77.44</td>
<td>-</td>
</tr>
<tr>
<td>ResNet-18 + ICASC${}_{A, ch}$</td>
<td><strong>0.981</strong></td>
<td><strong>83.17</strong></td>
<td><strong>5.73</strong></td>
</tr>
</tbody>
</table>

Multi-class Image Classification

We conduct multi-class image classification on the PASCAL VOC 2012 dataset, which contains 20 classes. Different from the above generic and fine-grained image classification where each image is associated with one class label, for each of the 20 classes, the model predicts the probability of the presence of an instance of that class in the test image. As our attention is class-specific, we can seamlessly adapt our pipeline from single-label to multi-label classification. Specifically, we apply the one-hot encoding to corresponding dimensions in the predicted score vector and compute gradients to generate the attention for multiple classes. As for the most confusing class, we consistently determine it as the non-ground truth class with the highest classification probability.

For evaluation, we report the Average Precision (AP) from the PASCAL Evaluation Server [11]. We also compute the AUC score via scikit-learn python module [51] as an additional evaluation metric [52]. Table 2.8 shows that ResNet-18 [2] with $A_{ch}$ outperforms the baseline by 5.73%.

Besides the mean Average Precision (mAP) shown in Table 2.8, we also provide the results for each category in Table 2.9. We notice that ResNet-18 guided by our ICASC${}_{A, ch}$ supervision gives the best performance in most of the categories, resulting in the best overall mAP score. When using Grad-CAM [9] as the attention guidance, the ICASC${}_{Grad-CAM}$ also outperforms the baseline method ResNet-18, which further validates the effectiveness of our proposed attention-driven learning framework ICASC.
Comparing Attention Mechanisms

We compare the image classification performance when ICASC is trained with Grad-CAM [9] and $A_{ch}$. As can be noted from the results in Table 2.4 and Table 2.10, the higher Top-1 accuracy of ICASC$_{A_{ch}}$ shows that our attention mechanism provides better supervisory signals for model training than Grad-CAM [9]. Additionally, even ICASC with Grad-CAM still outperforms the baseline, further validating our key contribution of attention-driven learning for reducing visual confusion. Our ICASC is flexible to be used with any existing attention mechanisms, resulting in improved classification performance.

We quantify attention overlap using histograms as shown in Figure 2.11a and Figure 2.11b. The x-axis shows the average attention overlap between the target and the top2 confusing classes, and the y-axis indicates the data proportion in each bin. Given attention
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>+ ICASC Grad−CAM</th>
<th>+ ICASC A&lt;sub&gt;ch&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pascal VOC 2012</td>
<td>77.44</td>
<td>82.12</td>
<td>83.17</td>
</tr>
<tr>
<td>CUB-200-2011</td>
<td>81.70</td>
<td>85.45</td>
<td>86.20</td>
</tr>
<tr>
<td>Caltech-256</td>
<td>80.01</td>
<td>80.28</td>
<td>81.32</td>
</tr>
<tr>
<td>ILSVRC2012</td>
<td>69.51</td>
<td>69.84</td>
<td>69.90</td>
</tr>
</tbody>
</table>

Table 2.10: Comparing baseline, ICASC<sub>Grad−CAM</sub> and ICASC<sub>A<sub>ch</sub></sub>.

Figure 2.11: The attention overlap histograms. ACC: Top-1 classification accuracy.

We experimented with ResNet-110 model on CIFAR-100 dataset, generating attention by Grad-CAM. Figure 2.11a quantifies that Grad-CAM gives overlapping attention in the last layer and inner-layer has less overlap. The last layer attention of the target class has a high chance of overlap with the confusing classes, with 0.66 pIoU in peak proportion. On the other hand, inner layer attention has less overlaps, having 0.39 pIoU in peak proportion. These results are consistent with our observations about inner and last layers, where inner layer attentions are more separable than the higher layer.
Figure 2.12: Improvements in top-1 predictions with our method (ResNet-18 + ICASC\textsubscript{A\textsubscript{ch}}) when compared to the baseline (ResNet-18). Top row: ResNet-18; bottom row: ResNet-18 + ICASC\textsubscript{A\textsubscript{ch}}.

Our goal is to improve model discriminability and generalizability, which we evaluate by means of image classification. The key insight is that there is attention overlap among various classes, which we refer to as visual confusion, and that eliminating the confusion with attention separability will improve model performance. This is further validated quantitatively by Figure 2.11b, where with the same backbone ResNet-110, ICASC results in better separability than the baseline, e.g. the peak pIoU value of 0.57 in ICASC and the baseline is with 0.66, leading to the higher Top-1 accuracy.

**Qualitative Results**

We show additional qualitative results for our proposed method in Figure 2.12 and Figure 2.13. Each figure shows four examples, where for each example, we show the input image and the ground-truth class in the first column, the top-5 categorical attention maps for the baseline in the top row of the adjacent columns, and those with our approach in the bottom row. In Figure 2.12, where the images are in high resolution, the baseline method is ResNet-18 and our method is ResNet-18 + ICASC\textsubscript{A\textsubscript{ch}}. In Figure 2.13, the baseline method is ResNet-110 and our method is ResNet-110 + ICASC\textsubscript{A\textsubscript{ch}}. In all the figures, the
Figure 2.13: Improvements in top-1 predictions with ResNet-110 + ICASC_{A,h} when compared to the baseline (ResNet-110). Top row: ResNet-110; bottom row: ResNet-110 + ICASC_{A,h}.

Ground-truth class attention map is marked using a red bounding box. There will be no marked attention map if the ground-truth class is not in the top-5 predictions. These figures show that our discriminative attention achieves better attention separability, with our model attending to regions that tell different categories apart. On the other hand, we observe visual confusion with the baseline, with high responses in the attention maps located at similar spatial locations among different categories.

Since discriminative attention is our principled learning objective, attention responses given by our method across the top-5 categories are more separable than those from the baseline method, and our trained model is able to attend to semantically discriminative parts of the ground-truth objects, resulting in the better classification results. For example, in the top left “cake” example in Figure 2.12, for both “cake” and “fried egg,” the baseline method attends to the central areas, containing the fruits and the cream around, which leads to visual confusion and misclassification of the image as “fried egg,” whereas our method attends to the central part (fruits and cream) for “cake” and the right part (cream) for “fried egg,” classifying the image as “cake” correctly. Additionally, in Figure 2.13, our method brings the ground-truth class to the top-1 which is out of top-5 predictions in the baseline method.
2.5 Conclusion

We propose a new framework, ICASC, which makes class-discriminative attention a principled part of training a CNN for image classification. Our proposed attention separation loss and attention consistency loss provide supervisory signals during training, resulting in improved model discriminability and reduced visual confusion. Additionally, our proposed channel-weighted attention has better class discriminability and cross-layer consistency than existing methods (e.g. Grad-CAM [9]). ICASC is applicable to any trainable network without changing the architecture, giving an end-to-end solution to reduce visual confusion. ICASC achieves performance improvements on various medium-scale, large-scale, fine-grained, and multi-class classification tasks. While we select last two feature layers which contain most semantic information to generate the attention maps, ICASC is flexible w.r.t. layer choices for attention generation, and we plan to study the impact of various layer choices in the future.
CHAPTER 3
AN COUPLED ENCODER-DECODER NETWORK FOR JOINT FACE DETECTION AND LANDMARK LOCALIZATION

3.1 Introduction

Face detection has been one of most important and still open problems in Computer Vision and Human-Computer Interaction. Face landmark localization is a prerequisite for many facial analysis applications, such as face recognition, face modeling, and expression transfer. Many effective face detection and landmark localization algorithms have been proposed to close the gap from realistic situations [53, 54, 55, 56, 57]. However, face-in-the-wild conditions, such as large pose variation and occlusions, largely degrade the performance of the methods.

Rooting back to the seminal works, Viola-Jones face detector [58] and Active Shape Model [59] have achieved wide applications in their own fields. Many representative works have been proposed to expand and better interpret the above models, such as Deformable Part Models (DPM) [60] for face detection and Active Appearance Models (AAM) [61, 62] for landmark localization. However, seldom efforts are put on jointly dealing with the two problems. A well-established pipeline is that face landmark localization accepts the bounding boxes input from face detection. However, such rule is not necessarily to hold. Zhu and Ramanan [60] presented a pioneering work to jointly detect faces and facial key points using DPM. The internal correlation between facial key points and the overall face location is well captured by the deformable part model. The method is limited by its hand-crafted feature (HOG) and a predefined tree structure, which is a hard constraint and lacks flexibility in capturing the shape variations.

With the strong power of feature representation, convolutional neural networks (CNN)
Figure 3.1: An example of our coupled encoder-decoder framework result, simultaneously predicting the face regions (white bounding boxes) and the face fiducial points (white dots). The false positive responses for landmark localization are effectively suppressed by the coupled face detection task, in which regions marked by the red dash bounding boxes are classified as non-face.

have shown significant advantages over traditional methods in many fields, i.e., object detection [63] and semantic segmentation [64, 65]. Among those methods, the Faster-RCNN [63] demonstrates superior performance across almost all the detection tasks, i.e., ImageNet, PASCAL and KITTI, where the ROI-pooling for region proposals is a key factor to achieve fast sampling and high accuracy. In face detection, the cascaded CNN [6] employs multi-scale shallow network cascade to fast localize faces. The early rejection of false alarms on smaller scale speeds up the run-time significantly. The Faster-RCNN relies firmly on the output of region proposal networks (RPN), which needs to carefully design the anchors with different aspect ratios, and to prepare unbiased training data. The cascaded CNN achieves good efficiency but the shallow structure prevents the further improvement of effectiveness.

Sliding window on feature maps is a simple and natural improvement to the RPN, whereas the region classification net (RCN) remains the same. The ROI pooling fully supports the sliding window operation on feature maps. To seek discriminative feature maps, we investigate the well established encoder-decoder framework originated from semantic segmentation [64]. The encoder-decoder generates a set of feature maps in
different scales. As we observed, the feature maps not only capture the responses for face landmark localization but also for facial regions, as shown in Fig. Figure 3.1. The strong face region responses are always coupled with the strong responses of the landmarks. If we consider landmark localization as a sparse segmentation problem (classifying the landmark regions as foreground), the localization task becomes bounding box independent and the feature maps could be re-utilized for face detection.

As a consequence, we propose a novel coupled encoder-decoder network to simultaneously localize landmarks and detect faces. First, an encoder-decoder network followed by a shallow landmark regression network is set up end-to-end, where the feature maps from the intermediate convolutional layers are gathered. Second, ROI pooling is applied on the gathered feature maps in three scales to further extract features for the face region classification. Different from cascaded CNN [6], we apply sliding windows on feature maps instead of raw images. During training, the encoder-decoder and landmark regression are updated for one iteration, while the facial region classification is updated for another iteration. The alternative training is similar to the Faster-RCNN training of RPN and RCN, which achieves stable convergence.

3.2 Related Work

**Face Detection:** Early works on face detection focus on the handcrafted features and the face classifiers, i.e., the Viola-Jones detector utilizes the Haar feature combined with the Adaboost classifier [58], a vanilla DPM [60] was proposed to defend the model based methods with top performance and a template based classifier was proposed as well [57]. Different from model-based methods, Shen et al. [66] propose to detect faces by image retrieval. Li et al. [67] further improve it to a boosted exemplar-based face detector. As the development of deep convolutional neural networks (CNN), there are many successful CNN-based methods with much better performance, i.e., the Cascaded CNN [6] applies the cascade of multi-resolution shallow networks to detect faces. Rather than training each
cascade stage independently, in [56], authors propose a joint training framework to learn the cascade model. The Convolutional Channel Feature [68] fully utilizes the rich features from the convolutional layers of different channels. Farfade et al. [69] proposed the multi-view deep neural network based framework to detect faces. Several state-of-the-art methods demonstrate the advantages of deep neural networks. [70] presents a method of end-to-end integration of a ConvNet and a 3D model for face detection in the wild. [71] used a CNN to detect facial parts and combine parts for holistic face detection, Ranjan et al. fuses the DPM with a deep pyramid structure [72]. [73] takes advantage of CNN futures and proposes an effective framework for finding small faces, demonstrating that both large context and scale-variant representations are crucial. [74] introduces the Single Stage Headless (SSH) face detector that, unlike two-stage proposal/classification approaches, detects faces in a single stage.

**Face Landmark Localization:** The model-based methods are back-traced to the Active Shape Models [59] and Active Appearance Models [61, 62]. Tons of improvements have been proposed, such as Constrained Local Model [75, 76], probabilistic matching [77], DPM [60], etc. The regression based landmark localization [78, 79, 80, 81, 82, 83, 84, 85] significantly improve the performance and run-time. In [85], multiple cascaded regressors are proposed with the capability to handle global shape variation and irregular appearance-shape relation. Those regression-based methods directly regress landmark locations from the features, reducing the complexity of model update. However, regression-based method is sensitive to the initial bounding boxes and is feature-specific where regression embedding firmly relies on the feature representations. Recently, the CNN based approaches show more compelling performance than regression. [53] proposes to use three stages of neural networks to cooperatively localize facial landmarks. [86] applies coarse-to-fine auto-encoders for the regression of landmark positions. In [87], a lightweight and compact CNN architecture is designed for landmark localization. [88] introduces a Recurrent Attentive Refinement network for facial landmark regression where landmark locations are refined progressively.
Figure 3.2: The Illustration of coupled encoder and decoder network. (a) illustrates the encoder and decoder layers ($f_{ENC}$ and $f_{DENC}$), which consists of convolution(Conv), max pooling, up-sample pooling and fully convolutional layers. (b) shows the coupling structure in which we collect the feature maps from $f_{ENC}$ and $f_{DENC}$ for procedure (c). (c) shows the coupled cascade face detection net $f_{CLS}$, where the sliding window and ROI pooling are applied on the feature maps to generate the feature representations. The proposals, classified as positive (green bounding boxes) in all three stages, are collected and non-maximum suppressed as face regions.

In [54], the multitask training strongly suggests joint dealing with multiple jobs while boosting each task’s performance. [55] proposes the shape basis network to fast approach the global optimal and point transformer network to refine the local shape variations. Compared to the efforts which explicitly import cascade structures, we propose a unified encoder-decoder model to incorporate both face detection and landmark localization, which boosts the learning convergence of the feature maps and the localization accuracy.

**Joint Face Detection and Landmark Localization:** As the first work that jointly handles face detection, landmark localization, Zhu and Ramanan [60] proposed a DPM based framework and achieved promising results in face-in-the-wild conditions. The similar structure is applied by [89] to detect and localize faces under occlusion. [89] claims to
simultaneously achieve the two tasks as well. However, it is more of a joint framework of using DPM to achieve face detection and landmark localization, which lacks consideration of the interactive boosting between the two tasks. The cascaded face detection and alignment [90] jointly deals with the two tasks, where it actually regresses the landmark positions after the face detection. Under the multi-task learning frameworks, several CNN based methods are recently proposed, i.e. [91] applied a cascaded CNN for multi-task learning, [92] integrated many tasks, face detection, landmark localization, pose estimation and gender recognition.

Encoder-Decoder Networks: The encoder and decoder networks are well studied in machine translation [93], where the encoder learns intermediate representation and the decoder transforms that representation. It is intensively investigated in speech recognition [94] and computer vision [64, 65, 95, 96]. In [96], the encoder-decoder architecture is applied to estimate human pose. In [64], authors applied an encoder-decoder structure on the semantic segmentation. The proposed algorithm in [64] mitigates the limitations of the previous methods based on fully convolutional networks by integrating deconvolutional network and pixel-wise prediction, which identifies detailed structures and handles objects in multiple scales. In this work, we employ the encoder-decoder network to learn the discriminative features for describing faces which can be shared by face detection and landmark localization. The architecture differs from the parallel multi-task framework [54] in which convolutional layers are shared and the last fully connected layers are split according to different tasks.

3.3 Coupled Encoder-Decoder Network

We propose the coupled encoder-decoder network in a unified framework which consists of three modules: 1) an encoder-decoder to predict facial response maps; 2) a coupled cascade face detection network sharing the feature maps with the encoder-decoder; 3) a regression network that outputs the 2D coordinates of facial landmarks.
3.3.1 The Encoder-Decoder for Facial Response Map Prediction

Semantically, landmark localization is a sparse segmentation problem. Segmenting the landmark regions are feasible without the constraint of the bounding boxes. As the encoder-decoder framework has shown strong evidence in the performance of segmentation [64], we employ it as the facial response map provider.

The network in Fig. Figure 3.2(a) takes an image $I \in \mathbb{R}^{w \times h \times 3}$ as input and a corresponding label map $Z \in \mathbb{R}^{w \times h \times 1}$ as ground truth. Each pixel in $Z$ is a discrete label $\{0, 1, 255\}$ that marks the presence of facial landmarks, where 0 denotes a non-landmark region, 1 for landmark and 255 set as ignore label for uncertain areas.

The encoder incorporates a set of convolutional layers, pooling layers and batch normalization layers [97], which is to encode the input $I$ into a feature space $C$:

$$C = f_{ENC}(I; \theta_{ENC}), C \in \mathbb{R}^{w_c \times h_c \times d_c} \quad (3.1)$$

where $C$ denotes the encoded $w_c \times h_c \times d_c$ feature maps and $\theta_{ENC}$ denotes encoder parameters. Symmetrically, the decoder module involves a set of unpooling, convolution and batch normalization to transform the feature maps $C$ to the 2-channel response maps $M$ in the same size of the image:

$$M = f_{DENC}(C; \theta_{DENC}), M \in \mathbb{R}^{w \times h \times 2} \quad (3.2)$$

where $\theta_{DENC}$ denotes the decoder parameters. The objective is formulated as a pixel-wise two-class classification problem with cross-entropy loss:

$$\mathcal{L}^{map} = \frac{1}{N_{px}} \sum_{i=1}^{N_{px}} y_{i}^{m} \log(p_{i}^{m}) + (1 - y_{i}^{m})\log(1 - p_{i}^{m}) \quad (3.3)$$

where $N_{px}$ denotes the number of pixels (ignored pixels are excluded); $p_{i}^{m} = g(f_{DENC})$ is
the probability of \( i \)-th pixel belonging to the landmark region and \( y_i^{m} \) is the ground-truth.

The response map \( \mathbf{M} \) plays a significant role in the whole framework for two reasons. First, it provides the attention maps of the foreground (fiducial point regions) and background. A shallow regression model is able to regress the coordinates in favor of the spatial information preserved by the encoder-decoder. Second, \( \mathbf{M} \) also provides the facial region information, which can be re-utilized for the face detection.

### 3.3.2 The Coupled Feature Map Cascade for Face Detection

In [6], authors propose a cascade framework consisting of 12, 24 and 48-nets, which early rejects non-face regions in the lower scale net (12-net) and passes the detected proposals to networks in the larger scale (24 and 48-nets) for aggregation. However, sliding window on the original image is time-consuming, which may restrict [6] to adopt deeper networks and higher image resolution. As in Figure 3.2 (b), the same scale intermediate feature maps from \( f_{ENC} \) and \( f_{DENC} \) are concatenated as the feature maps for face detection. The sliding window is applied to the feature maps instead of original images, avoiding redundant convolutional computation. ROI pooling [63] is applied to map each sub-region of feature maps into a feature vector in the fixed dimension.

The learning objective is formulated as a binary classification as well as the bounding box coordinates localization.

\[
\mathcal{L}^{det} = \mathcal{L}^{cls} + \gamma \mathcal{L}^{loc} \\
\mathcal{L}^{cls} = y_i^{cls} \log(p_i^f) + (1 - y_i^{cls})(1 - \log(p_i^f)) \\
\mathcal{L}^{loc} = \| (x_1, x_2) - (x_1^*, x_2^*) \|_2^2
\]

where the classification loss \( \mathcal{L}^{cls} \) is defined as the cross-entropy loss over the probability \( p_i^f \) of the \( i \)-th window being a face and \( y_i^{cls} \in \{0, 1\} \) denotes the ground-truth. The loss of bounding box localization \( \mathcal{L}^{loc} \) is defined as the euclidean distance between the ground
Figure 3.3: The architecture of $f_{REG}$. ROI pooling is applied on the foreground channel and the last encoder’s feature layer, yielding feature vectors $f_i$ and $f_g$. $f_{REG}$ utilizes the concatenation of $f_i$ and $f_g$ to predict 2D coordinates for $N$ landmarks.

truth bounding box denoted as $(x_1^*, x_2^*)$ upper-left and bottom-right corner points and the predicted two points $(x_1, x_2)$. The regularization factor $\gamma$ is set up to balance the penalty from the two branches. We apply 0.01 as the typical value in our framework.

### 3.3.3 Landmark Localization From Response Maps

As shown in Figure 3.3, the model $f_{REG}$ combines the response maps $M$ and feature maps in the last layer of $f_{ENC}$ to predict landmark coordinates. Only the foreground channel of response map $M$ is used for landmark localization. According to the detected window, ROI pooling is applied on the foreground channel and on the last encoder’s feature layer, yielding feature vectors $f_i$ and $f_g$ respectively. We concatenate $f_i$ and $f_g$ for landmark localization, taking advantage of both local and global information. The concatenated feature is fed to a shallow regression network in which the last layer is a fully connected layer with $2N \times 1$ neurons, which outputs 2D coordinates of $N$ facial key points. The landmark localization is formulated as a regression problem with Euclidean loss:

$$L_{reg} = \sum_{i=1}^{N} \left( \frac{(sx_i - sx^g_i)}{w} \right)^2 + \left( \frac{(sy_i - sy^g_i)}{h} \right)^2$$

(3.5)

where $(sx, sy)$ is the coordinate of detected facial points and $(sx^g, sy^g)$ is the ground truth. The distance in $x$ and $y$-axis is normalized by window width $w$ and height $h$, respectively.
3.4 Implementation Details

In this section, we describe the architectures and training procedure for the three proposed modules.

3.4.1 Network Architectures

*Encoder $f_{ENC}$ and Decoder $f_{DENC}$*

The encoder is designed based on a modification of the VGG-16 network [98]. There are 13 convolutional layers with $3 \times 3$ filters corresponding to the first 13 convolutional layers in VGG-16. The fully connected layers are replaced by fully convolutional layers, which preserve spatial information. The $f_{ENC}$ contains 5 max-pooling layers in $2 \times 2$ size and a constant stride of 2. A 2-bit code strategy introduced by [99] is applied to record the spatial information of the maximum activation. At the corresponding unpooling layer, such spatial information is utilized to recover each activation back to its original location. The $f_{DENC}$ is in a mirrored configuration of the $f_{ENC}$ except replacing max-pooling with unpooling layers. The decoder outputs a 2-channel response map which is fed to a softmax classifier to predict pixel-wise confidence. Batch normalization [97] and rectified linear unit (ReLU) [100] are applied after each convolutional layer to reduce internal shift within a mini batch.

*Coupled Face Detection $f_{CLS}$*

Figure 3.2 (b) demonstrates that the feature maps from both encoder and decoder in the same scale are concatenated, which occurs in three scales: 1) $Conv_{2.2}$ and $Deconv_{2.2}$; 2) $Conv_{3.3}$ and $Deconv_{3.3}$; 3) $Conv_{4.3}$ and $Deconv_{4.3}$. In 1), face detection begins with dense scanning over the feature maps. In this sale, scanning by a $5 \times 5$ window with 1-pixel stride is equivalent to a $40 \times 40$ window with stride of 8 on the original image, obtaining $\left\lfloor \frac{(W - 40)}{8} \right\rfloor + 1 \times \left\lfloor \frac{(H - 40)}{8} \right\rfloor + 1$ candidates. ROI pooling is applied to map each window to a 256-d feature vector (128-d for $Conv_{4.3}$ and $Deconv_{4.3}$ respectively).
The feature vector is fed into fully connected layers of 128 neurons followed by a softmax classifier, generating confidence score for the specific window. A threshold $T_1$ is set to reject non-face areas. NMS is applied on highly overlapped proposals to reduce the output windows. In the second scale, the ROI pooling transforms regions preserved in previous stage into a 512-d feature vector. A threshold $T_2$ is set to filter out non-face regions further. In the last scale, we continue to examine the preserved windows. The dimension of feature vectors generated by ROI pooling is 512 and fully connected layers have 256 neurons. A threshold $T_3$ is set to reject false alarms. The three stage box calibration networks introduced by [6] are removed as the spatial alignment is naturally incorporated by the feature maps.

**Landmark Regression $f_{\text{REG}}$**

As shown in Figure 3.3, the $f_{\text{REG}}$ applies fully connected layers of 512 neurons to directly regress the input to the 2D coordinates. The network input is a combination of the response map $M$ and the feature maps given by $f_{\text{ENC}}$. According to detected windows, ROI pooling transforms each region of interest into a feature vector with dimension of 512. The dropout layer with 0.5 probability is also applied. In our task, $N$ number of landmarks is set as 7.

### 3.4.2 Training

In our framework, we apply an alternative training procedure for the coupled structure. First, the model of $f_{\text{ENC}}$ and $f_{\text{DENC}}$ followed by the $f_{\text{REG}}$ are trained end-to-end, in which the gradients could be back-propagated without any gradient interception. Then, the coupled feature maps are concatenated. ROI pooling is applied on the feature maps to generate features for the facial region classification and bounding box localization. In the last, $f_{\text{REG}}$ is fined tuned with cropped sample images according to the windows given by $f_{\text{CLS}}$. The first step is considered the mainstream, while the second step is based on the feature maps generated in the first one. By alternatively optimize each part, the two objectives are optimized simultaneously.
In the first step, the convolutional parameters are initialized by weights of VGG-16 trained on large datasets for object classification. The rest parameters are set with Gaussian Distribution. In this step, the data augmentation is performed, including horizontal flip, central rotation ($\pm 10^\circ$) and scaling ($0.8 - 1.2$), yielding 24 variations for one image.

The second step involves a cascade of the three stages of $f_{CLS}$. We crop patches by sliding window to collect positive and negative samples to train the first stage. Patches of Intersection-of-Union (IoU) larger than 0.6 to ground truth are labeled as positive. The negative samples are regions of IoU less than 0.2. Additionally, we add more negative samples by collecting around 2000 background images, from which we randomly sample 100,000 non-face patches. The detector of the first stage is applied to mine positive and negative samples for the second stage. The non-face regions with confidence score given by the detector higher than threshold $T_1$ become negative samples. Similarly, the detectors of the first and second stages are both used to mine the training samples for the third stage.

The face detection follows the cascaded structure in [6]. $T_1$ in the first stage is set to keep 98% recall on the validation set, which rejects 85% false positive windows. Threshold $T_2$ in the second stage is set to keep 95% recall on the validation set.

Sliding window is applied to feature maps with various sizes for multi-scale detection, e.g., a 5x5 or 10x10 window with 1-pixel stride on $Conv_{3,3}$ feature map is equivalent to 40x40 or 80x80 window on original image with stride of 8.

The last is a fine-tuned step for $f_{REG}$ where training samples are the cropped images according to the windows given by $f_{CLS}$. In this step, $f_{REG}$ shares the feature maps with $f_{ENC}$ and $f_{DENC}$. In the training stage, given image size of 224x224 and batch size of 64, one iteration of $f_{ENC}$ and $f_{DENC}$ takes 0.3s, and $f_{CLS}$ or $f_{REG}$ takes 1s.
3.5 Experiments

3.5.1 Experimental Setup

For landmark localization, the training data consists of images from training set of LFPW [101] (LFPW-train) and Helen [102] (Helen-train). The evaluation set contains AFW [60], testing set of LFPW (LFPW-test) and Helen (Helen-test). We follow the annotation rule in [103] for 68 facial points to generate 7 landmarks to locate eye corners, mouth corners and nose tip. The facial images used in our experiments cover large head pose variations, expressions, variations of background and occlusions.

For face detection, we apply a commonly used wild face dataset, WIDER FACE [104], for training set. The testing is conducted on two mostly deployed public benchmarks, FDDB [105] and AFW [60]. WIDER FACE consists of 393,703 labeled face bounding boxes in 32,203 images. FDDB dataset contains the annotations for 5,171 faces in a set of 2,845 images and AFW [60] is a 205-image dataset with 468 faces annotated. Images of the three datasets contain cluttered backgrounds and large variations in viewpoints and appearance.

3.5.2 Evaluation of Face Landmark Localization

We first evaluate the coupled encoder-decoder network for face landmark localization. The localization accuracy is measured by the pixel distance between detected points and the ground truth. We follow the evaluation metric in [54] where the pixel distance is normalized by the inter-ocular distance. As illustrated in Fig. Figure 3.4, the performance of our model is compared with four methods including 1) SDM [79]; 2) DLIB [83, 106]; 3) TCDCN [54]; 4) CoR [78]; 5) HPM [89].

The bottom row of Figure 3.4 illustrates statistical curves of mean localization errors on three datasets, Helen-test (left column), AFW (middle column) and LFPW-test (right column). According to the bottom row of Figure 3.4, the accuracy of our approach is better than
Figure 3.4: The face landmark localization results on three benchmarks, Helen (left column), AFW (middle column) and LFPW (right column). The top row illustrates the localization errors for seven facial components, left corner of left eye (le_l), right corner of left eye (le_rc), left corner of right eye (re_l), right corner of right eye (re_rc), mouth left corner (m_l), mouth right corner (m_r) and nose tip. The bottom row shows the cumulative curves of relative mean errors, where the horizontal axis is the normalized distance with respect to the inter-ocular distance and the vertical axis is the proportion of images in the dataset.

The other four methods on datasets of Helen-test and LFPW-test. Regarding of AFW, our coupled encoder and decoder model is comparable to DLIB, but still better than other three.

The top row of Figure 3.4 shows the localization errors for the seven facial components. According to the two histograms of Helen-test and AFW, the accuracy of our approach is comparable to DLIB with respect to left eye’s left corner and right eye’s right corner. In the LFPW-test set, our approach is comparable to SDM for right eye’s right corner and DLIB for left eye’s right corner. Regarding the rest facial components, the localization accuracy of our approach is higher than the four methods. The localization accuracy of our coupled encoder-decoder network for mouth corners and nose tip is higher than other methods by a significant margin.

The evaluation conducted on the three benchmarks demonstrates the superior performance of our coupled encoder-decoder network than the state-of-the-art methods. It can be interpreted as the fact that our model captures the more discriminative features and the segmentation scheme is more effective than the regression based methods. With the powerful
Figure 3.5: The evaluation of face detection on datasets of FDDB (left) and AFW (right). On the FDDB dataset we compare our performance with the state-of-the-art methods including: CascadeCNN [6], Joint Cascade [90], DDFD [69], ACF-multiscale [107], PEP-Adapt [108], Boosted Exemplar [67], HeadHunter [57], Pico [109], Viola-Jones [58], TSM [60], HPM [89]; On the AFW dataset, comparison methods includes: HeadHunter [57], DPM [57], SquaresChnFtrs-5 [57], Shen et al. [66], TSM [60], and three commercial applications, Face++, face.com and Piscasa.

3.5.3 Evaluation of Face Detection

We compare our coupled face detector with the state-of-the-art approaches on FDDB and AFW benchmarks. For FDDB, we compare our performance directly with the published methods listed in FDDB platform [105]. Two evaluation protocols are provided by [105], discontinuous score and continues score. Continuous score heavily relies on annotations of training set. We do not follow the eclipse labelling style for the faces, so we only report discontinuous score, where detected regions of IoU larger than 0.5 to the ground truth are regarded as true positives. For AFW, we use the toolbox provided by [57] to evaluate the detection performance.

The evaluation on FDDB and AFW is illustrated in Figure 3.5. In both datasets, our performances are favorably comparable to the state-of-art methods. More worth to highlight, solving the same problem as ours, both Joint Cascade [90], TSM [60] and HPM [89] are
proposed to jointly detect faces and localize landmarks. Our approach achieves higher accuracy than the two methods on FDDB. While on AFW, our detection accuracy is consistently higher than TSM by a significant margin. Even without box calibration networks, our structure demonstrates the better performance than the previous cascade CNN [6] on FDDB. Two main reasons may lead to the performance boosting: 1) the features captured by the coupled encoder-decoder are more discriminative for describing faces; 2) our approach takes advantage of multi-task training which brings mutual benefits among different tasks.

Fig. Figure 3.6 shows that there are several faces miss detected in both benchmarks. For face detection as a single task, in order to achieve top performance, the detectors are designed to be able to find face regions even in small size, low quality and heavy occlusion, as depicted in Figure 3.6. Our approach tries to jointly detect faces and localize the landmarks, where the encoder-decoder are trained with faces to be reasonable clear and in prosper size, without covering such extreme cases. The miss rate indicates our future effort can be put on exploration of dealing with the facial images in low quality.

Robust Pre-processing for Face Recognition

In the study [110], a Face Recognition(FR) dataset is proposed where faces are collected with multi-sensor mobile devices in challenge conditions. The challenges for mobile based FR are variation in illumination conditions, poor face image quality (due to various factors
Figure 3.7: Face examples of the multi-sensor database, which are collected using various cell phones and photographed in different distances.

including noise and blurriness due to movement of hand-held device during collection), variations in face pose and camera sensor quality. Those in-the-wild conditions make face area localization inaccurate, which causes recognition performance degrade. In this section, we apply the $f_{CLS}$ module as the FR pre-processing. The detection accuracy up to 98.4% of detecting faces within 1 meters distance to the sensor demonstrates the robustness of our approach to handle cases in the mobile condition.

In [110], the multi-sensor (MS) face image database is collected using a set of cell phone devices including Samsung S4 Zoom, Nokia 1020, Samsung S5 and iPhone 5S. The visible band face database is collected indoors, outdoors, at standoff distances of 1m, 5m and 10m respectively, and with different pose angles as shown in Figure 3.7.

We conducted the experiments for faces photographed in all the three distance settings, 3459 frames are sampled for each. In total, we uni-sampled 10377 frames from the videos and processed the images using $f_{CLS}$. The number of successful and failure cases are 8860 and 1517, respectively. The success rate is up to 85.4% and 98.4% for 1m setting:

(i) the numbers of failure cases are 54, 360, 1103 for distance settings of 1m, 5m and
(ii) most of the failure cases are people photographed in 10m and 5 meters; it is due to missing the faces which are in the uneven/low illumination (Figure 3.8 left) or blur (Figure 3.8 right), and tiny (photographed in long distance);

(iii) the failure cases of close faces (1m) are caused by sampling the dark frames and camera pointing to wrong direction where the faces are not or partially shown in the frames.

The results of applying the $f_{CLS}$ module to detect the faces photographed by mobile devices demonstrate the robustness of our approach as to be the pre-processing step for Face Recognition. In FR, users are most likely to present their faces closed to the sensor and $f_{CLS}$ finds all the visible faces in the case of short distance.

### 3.5.4 Qualitative Results

Figure 3.9 shows qualitative results of joint face detection and keypoints localization, which are performed simultaneously under deep neural network frameworks. Different from other landmark localization methods, we are able to localize landmarks without a face bounding box prior and generate the 7-point landmarks. In our work, we aim to generate the semantic feature maps to boost the performance of face detection and landmark localization. By
Figure 3.9: Qualitative results of face detection and landmark localization.

carefully defining the landmark positions on top of the feature maps, the landmarks could have semantic meanings, i.e., the 7-point setup as denoting the eye centers, nose tip and the mouth corners. If we adopt the 68-point annotation as the other landmarks settings, some landmarks such as along the profile may be less meaningful.

In our work, the two tasks would boost each other in generating better features. The encoder-decoder framework is proposed to generate the feature map, which indicates semantic facial structures as shown in Figure 3.1 the feature attention maps. This semantic highlight is also observed in face detection. Thus, setting up a coupled structure for face detection and landmark localization, the response map constrained from face detection would be also beneficial for the landmarks.

3.6 Conclusion

In this paper, we proposed a coupled encoder-decoder neural network to jointly detect faces and localize landmarks. The encoder-decoder provides the discriminative feature maps for landmark localization. Further, we observe that the feature maps is also effective for the task of face detection, which enables a unified coupled structure as proposed in our
method. The performance on both of the two tasks are very competitive while sometimes better than some of the state-of-the-art methods. The training of the overall framework is alternative optimization. Future work will focus on how to formulate the two tasks as a single optimization problem.
CHAPTER 4
LEARNING TRAILER MOMENTS IN FULL-LENGTH MOVIES WITH
CO-CONTRASTIVE ATTENTION

4.1 Introduction

“Just give me five great moments and I can sell that movie.” – Irving Thalberg (Hollywood’s first great movie producer).

Movie is made of moments [111], while not all of the moments are equally important. In the spirit of the quote above, some key moments are known as coming attraction or preview, which can not only grab an audience’s attention but also convey the movie’s theme.

The importance of detecting the key moments is two-fold. First, key moments migrate the content overwhelming. There are millions of movies produced in human history [112]. A full-length movie typically lasts two or three hours, making it incredibly time-consuming for consumers to go through many of them. The key moments in the form of short video clips can make the movie browsing efficient, where audiences can quickly get the theme by previewing those short clips with story highlightings. Second, for the purpose of movie promotion, the well-selected moments can attract audience to the movie, where the key moments are usually drawn from the most exciting, funny, or otherwise noteworthy parts of the film but in abbreviated form and usually without spoilers\(^1\).

A popular form of key moments in the movie industry is the trailer, which is a short preview of the full-length movie and contains the significant shots selected by professionals in the field of cinematography. In this paper, we focus on moments in the movie trailer and try to answer an important question regarding Movie Trailer Moment Detection (MTMD) – can we learn a vision model to detect trailer moments in full-length movies automatically?

\(^1\)https://en.wikipedia.org/wiki/Trailer_(promotion)
Figure 4.1: We leverage the trailer shots to estimate the attention scores of individual shots in the full-length movie, which indicate the “trailerness” of the shots and can be used as weak supervision to model the contrastive relation between the key and non-key moments in the feature space.

The MTMD problem is related to the existing line of research on Video Highlight Detection (VHD), a task of extracting highlight clips from videos. Recently, deep learning has become a dominant approach to this task, which formulates it as a problem of learning a ranking model to score the human-labeled highlight clips higher than the non-highlight. Given video clips, the deep spatial-temporal features are extracted as the input to train the ranking model [113, 114, 115, 116, 117]. However, the existing VHD approaches cannot be directly applied to MTMD due to the following reasons.

First, there is no labeled data available for MTMD. To train a robust VHD model, it requires extensive supervision where the annotators must manually identify the highlight clips. Though few efforts have been made to conduct unsupervised VHD, their inferior performance below the supervised indicates the requirement for supervision. It seems reasonable to annotate the highlights which demonstrate specific actions (e.g., “Skiing”, “Skating”) or events (e.g., “Making Sandwich”, “Dog Show”) as in the VHD datasets like Youtube Highlight [118] and TVSum [119]. However, annotating trailer moments in movies is much more challenging as the selection of trailer moments might attribute to various factors such as emotion, environment, story-line, or visual effects, which requires the annotators to have specialized domain knowledge. To resolve this issue, we create the
supervision signal by matching moments between the trailers and the corresponding movies, as shown in Figure 4.1. Specifically, we propose a Co-Attention module to measure the coherence between the shots from trailers and movies, through which a set of the best and worst matched shots from the movies are discovered as weakly labeled positive and negative samples.

Second, the existing VHD approaches treat the individual short clips in the long videos separately without exploring their relations. In fact, the trailer moments follow certain common patterns and should be distinguishable from the non-trailer moments. Taking action movies as an example, although different movies tell different stories, their trailer moments always contain shots with intensive motion activities. To incorporate such prior into MTMD, we propose a Contrastive Attention module to enforce the feature representations of the trailer moments to be highly correlated while at the same time encourage the high contrast between the trailer and non-trailer moments. In this way, the features of trailer moments can form a compact clique in the feature space and stand out from the features of the non-trailer moments.

We integrate the two modules, i.e., Co-Attention and Contrastive Attention, into the state-of-the-art 3D CNN architecture that can be employed as a feature encoder with a scoring function to produce the ranking score for each shot in the movie. We dub the integrated network CCANet: Co-Contrastive Attention Network. To support this study and facilitate researches in this direction, we construct TMDD, a Trailer Moment Detection Dataset, which contains 150 movies and their official trailers. The total length of these videos is over 300 hours. We conduct experiments on TMDD, and our CCANet shows promising results, even outperforming the supervised approaches. We also demonstrate that our proposed Contrastive Attention module significantly achieves marginal performance-boosting over the state-of-the-art on the public VHD benchmarks, including Youtube Highlight [118] and TVSum [119].
4.2 Related Works

Studies on movie and trailer have been on the increase interests in computer vision research because of their rich content [112]. Several efforts have been made to analyze movies or trailers from different angles. A growing line of research is trying to understand the semantics in movies via audio-visual information together with the plot, subtitles, sentiment, and scripts [120, 121, 122, 123, 124, 125, 126, 127]. The works [120, 121, 122] focus on understanding the relationships of movie characters. Zhu et al. [128] proposed an approach to match movie shots and scripts so as to understand high-level storylines. Tapaswi et al. [123] developed a movie Q&A benchmark, proposing a way to understand movies via visual question answering. Chu et al. [124] use machine learning approaches to construct emotional arcs of the visual or audio signals, cluster the type of arcs, and predict audience engagement. Besides the studies on movies, there are also efforts trying to understand the trailers. The works in [125, 126] attempt to generate trailers for user-uploaded videos by learning from structures of movies. Smith et al. [127] present a heuristic system to fuse the multi-modality to select the candidate shots for trailer creation and the analysis is performed on horror movies. In [129, 130], the genre classification problem is investigated by using the trailers to represent the movie content. For this purpose, datasets with several thousand trailers have been constructed. These works are all based on the movie or trailers separately without considering their correspondence. As a pioneering work, Huang et al. [112] propose an approach to bridge trailers and movies, allowing the knowledge learned from trailers to be transferred to movie analysis. However, a dataset of full-length movies and the key moment annotations is still unavailable, which motivates us to collect TMDD.

Video highlight detection has been studied a lot for sports videos [131, 132, 133]. Recently, supervised video highlight detection has been applied to Internet videos [118] and first-person videos [134]. The Video2GIF approach [135] learns to construct a GIF for a video from the user-created GIF-Video pairs. The supervised highlight detection requires
human-labeled training pairs, which are expensive to obtain. Recently, several efforts have been made for unsupervised video highlight detection, which does not require manual annotations. These approaches can be further divided into domain-agnostic or domain-specific approaches. The domain-agnostic approaches operate uniformly on any video containing different semantic concepts. The approach in [136] is based on motion intensity. Works [137, 138] are to train a set of video category classifiers and then detect highlights based on the classifier scores or spatial-temporal gradients. In contrast, the domain-specific approaches train highlight detectors on a collection of videos containing the same concept. In [139], Yang et al. propose a category-aware reconstruction loss for unsupervised domain-specific highlight detection.

A very recent work [140] is proposed to get rid of human annotations by leveraging the video duration as the supervision. The key insight is that the clips from shorter user-generated videos are more likely to be the highlights than those from longer videos since users tend to be more focused on the content when capturing shorter videos [140]. While the insight does not apply to movie domain. As shown in Figure 4.2, the duration of the trailer and non-trailer shots is similar statistically, which severely mutes the duration signal.

Inspired by the fact that movies come with trailers, we tackle the annotation problem by leveraging the trailer moments to generate the supervision. A Co-Attention module is proposed to measure the coherence between the shots from trailers and movies. Different from the existing Pseudo-Label approach, which offline predicts the labels [141, 142], our Co-Attention module is updated in the learning process, where training is in an end-to-end fashion.
4.3 Approach

We develop CCANet with two goals: 1) with the weak-supervision from the publicly available trailers, the network is trained without human labeling; 2) we incorporate the “contrastive” relation into the learning process so that the trailer moment can be distinguishable from the non-trailer. We first describe how we construct the Trailer Moment Detection Dataset (TMDD) in Sec Subsection 4.3.1. Then we present the CCANet in Sec Subsection 4.3.2, consisting of the Co-Attention for learning the trailer moments and the Contrastive Attention for feature augmentation.

4.3.1 Trailer Moment Detection Dataset

We aim to detect the key moments in movies using the publicly available trailers as supervision. However, the existing movie or trailer related benchmarks [129, 130] are not appropriate for this task. They collect the trailers or the movie posters for genre classification without full movies provided. Recently, Huang et al.[112] learn the vision models from both movies and trailers by proposing a Large-Scale Movie and Trailer Dataset (LSMTD). However, LSMTD is not publicly available. Moreover, due to the different purposes of learning a semantic model for movie understanding, LSMTD has no ground-truth for MTMD evaluation. To this end, we construct a new dataset, named Trailer Moment Detection Dataset (TMDD).

TMDD contains 150 movies in full length paired with their official trailers. The movies are split into three domains according to the genre, including “Action”, “Drama”, and “Sci-Fi”. Each domain has 50 movie-trailer pairs. We train an MTMD model for each domain, which accounts for the intuition that the key moments are highly domain-dependent, e.g., a fighting moment might be crucial in “Action” movie but not in romantic “Drama”.

We define a movie moment as a shot that consists of consecutive frames in one camera recording time [143]. We apply the shot boundary detection [143] to segment movies and
trailers into multiple shots. Overall, the TMDD contains 263,837 movie shots and 15,790 trailer shots. Hence, MTMD on this dataset is a quite challenging task as the true positives only take \( \sim 6\% \) if we regard all trailer shots as the key moments. To our best knowledge, this is the first and largest dataset that has ever been built for MTMD.

To build the ground-truth without the requirement of experts annotating the key moments, we conduct visual similarity matching between trailers and movies at the shot-level and then manually verify the correctness of the matches. The shots occurring both in trailers and full-length movies are regarded as the ground-truth key moments in the movie. Notably, the annotations obtained in this way are only for performance evaluation but not for training the model. In the next section, we present our approach of leveraging the trailers to learn the movie key moments without human annotations needed.

### 4.3.2 CCANet for Trailer Moment Detection

We integrate the Co-Attention and Contrastive Attention modules into a unified CCANet, as shown in Figure 4.3(Left). Our goal is to learn a scoring function \( S(\cdot) \) that predicts the “trailerness” score of a movie shot given its feature as input, where the feature is extracted from the individual shot by a 3D ConvNet [144]. At test time, movie shots can be ranked based on the predicted scores, and the top-ranked shots are deemed as the key moments that can be applied to create trailers. Specifically, instead of relying on human annotations to create the pairwise shots for learning the \( S(\cdot) \), we create shot pairs based on the Co-Attention scores \( Att \) between trailers and movies. Additionally, the Contrastive Attention module is proposed to augment the 3D features so as to explore the relations between the trailer and non-trailer shots. The details are described below.

*Learning Trailer Moments via Co-Attention*

We leverage the Co-Attention between movies and trailers to modify the basic ranking loss for MTMD.
Figure 4.3: **Left:** overview of the proposed CCANet. We use the Co-Attention between the trailer and movie as the weak supervision and propose Contrastive Attention to augment the feature representations such that the trailer shots can stand out from the non-trailer shots in the feature space. **Right:** the details of Contrastive Attention module. ⨂ indicates matrix multiplication and “Concat” stands for vector concatenation.

**Basic Ranking Loss.** We assume that the movie dataset $D$ can be divided into two non-overlapping subsets $D = \{D^+, D^\}$, where $D^+$ contains the shots of key moments, $D^-$ contains the shots of non-key moment. Let $s_i$ refer to a movie shot and the 3D feature extracted from shot $s_i$ is $x_i$. Since our goal is to rank the shots of key moment higher than the shots of non-key moment, we construct training pairs $(s_i, s_j)$ such that $s_i \in D^+$ and $s_j \in D^-$. We denote the collection of training pairs as $P$. The learning objective is the ranking loss:

$$L_{\text{Rank}} = \sum_{(s_i, s_j) \in P} \max(0, 1 - S(x_i) + S(x_j)). \quad (4.1)$$

**Co-Attention between Trailer and Movie.** Let $T$ refers to a set of $N_t$ shots in a trailer. We encode each $t_i \in T$ into a 3D feature. As shown in Figure 4.3(Left), a linear layer is applied to map the shot features into a memory $M = \mathbb{R}^{N_t \times d}$, where $d$ is the dimension of the memory vector $m_\tau \in M$. Given the feature $x_i$ of shot $s_i$ from a full movie, we generate the query $q_i$ by applying the linear layer to $x_i$. The Co-Attention can be calculated as the maximal convolution activation between the query $q_i$ and the vectors in $M$:

$$\text{Att}_i = \max_{\tau \in N_t} (q_i \odot m_\tau). \quad (4.2)$$
The Co-Attention score $Att_i$ measures the coherence of shot $s_i$ in the movie to all shots in the trailer $T$. A large $Att_i$ value indicates that the shot $s_i$ is highly correlated to the trailer and therefore is a potential key moment in the movie.

**Ranking Loss with Co-Attention.** The ranking loss in Eq. (Equation 4.1) assumes that we have annotations for constructing the training set $D^+$ and $D^-$. However, it requires extensive human efforts and domain knowledge to annotate them. To achieve the learning goal without access to human annotations, we leverage the trailer to predict the attention score $Att_i$ and use it as a “soft label” to measure the importance of shot $s_i$ in the full movie. Additionally, as shown in Figure 4.3(Left), we introduce a Contrastive Attention module $g(\cdot)$ (described in the next section and illustrated by Figure 4.3(Right)) to augment the feature $x_i$ of shot $s_i$ into $f_i$. With the soft labels and augmented features, we can rewrite the learning objective as follows:

$$L_{Rank} = \sum_{(s_i, s_j) \in \mathcal{P}} w_{ij} \max \{0, 1 - \sigma[\mathcal{S}(f_i) - \mathcal{S}(f_j)]\}$$

where $w_{ij} = \lambda(\exp(|Att_i - Att_j|) - 1)$, 

$$\sigma = \text{sgn}(Att_i - Att_j),$$

(4.3)

where $\lambda$ is a scaling factor and $w_{ij}$ is introduced as a variable to identify the validness of a pair $(s_i, s_j) \in \mathcal{P}$ to the loss. The underlying intuition is that we assign a large weight to the contrastive pair where the difference between $Att_i$ and $Att_j$ is significant and therefore, should be treated as a confident training sample. The variable $\sigma$ is used to determine the order of the predicted scores based on their Co-Attention values.

It is worth noting that our approach module is different from the existing approach of *learning with Pseudo-Label* (PL). In PL, labels are collected offline from the highly confident predictions made by the model. While our Co-Attention module updates the label predictions in the end-to-end training process.
Augmenting Features via Contrastive Attention

As shown in Figure 4.3(Right), we draw inspiration from the attention mechanism [13] to exploit the contrastive relation among shots. Given a target shot $s_i$ and an auxiliary shot set $\tilde{S}$ with $N$ shots, we extract a 3D visual feature $x_i \in \mathbb{R}^d$ and a feature set $\tilde{X} \in \mathbb{R}^{N \times d}$, respectively. We apply $\tilde{X}$ as the supportive set to augment $x_i$ to be $f_i = g(x_i, \tilde{X}) \in \mathbb{R}^{2d}$.

We aim to make the attention contrastive such that the features of key moments can form a compact clique in the feature space and stand out from the features of the non-key moments. Specifically, the attention $A \in \mathbb{R}^{1 \times N}$ between $x_i$ and each $x_j \in \tilde{X}$ is computed as:

$$A(x_i, \tilde{X}) = \text{softmax} \left( \frac{a_i^T K}{\sqrt{d}} \right), \quad (4.4)$$

where we use linear layers to map $x_i$ and $\tilde{X}$ to a query vector $a_i$ and key matrix $K$ respectively, and $d$ is the output channel number of the linear layers. The attention score is used to weight the contribution of shots in $\tilde{S}$ to augmenting $s_i$. We apply another linear layer to map $\tilde{X}$ to a value matrix $V$. Then augmenting $x_i$ to be $f_i$ is formulated as:

$$f_i = \text{concat} \left[ x_i, \text{Linear(ReLU}(A(x_i, \tilde{X}) \cdot V)) \right]. \quad (4.5)$$

Now we describe how to construct the auxiliary shot set $\tilde{S}$ for a specific $s_i$ and how to regularize the feature augmentation discussed above. Inspired by our intuition that the cross-video key moments share common patterns and the key and non-key moments in the same video are supposed to be contrastive, we choose both common key moments and non-key moments to construct $\tilde{S}$. In particular, given a shot $s_i$ in a mini-batch during training, we collect all the key moment shots across videos as well as the non-key moment shots surrounding $s_i$ in the same video into $\tilde{S}$ (More details can be found in the supplementary material). The key and non-key moment shots in the supportive set $\tilde{S}$ are denoted by $\tilde{S}^+$ and $\tilde{S}^-$ respectively, and we propose the following loss as a regularizer to explicitly impose the contrastive relation between the key and non-key moments:

$$L_C = -\sum_i \theta_i \log \frac{\sum_{j \in \tilde{S}^+} \theta_j \exp(o_i^T k_j)}{\sum_{j \in \tilde{S}^+} \theta_j \exp(o_i^T k_j) + \sum_{j \in \tilde{S}^-} (1 - \theta_j) \exp(o_i^T k_j)} \quad (4.6)$$
where $k_j$ is the $j$-th vector in the embedding key matrix $K$ as in Eq. (Equation 4.4), and $\theta_i$ is a confidence weight indicating the reliability of the soft label for the shot, defined as a function of the Co-Attention score $Att_i$:

$$\theta_i = \frac{1}{1 + \exp(-\gamma(Att_i - \epsilon))}$$

where we empirically choose values of $\gamma$ and $\epsilon$ to be $0.65 \times \max(Att_i)$ and 100 respectively. Eq. (Equation 4.7) approximately maps the Co-Attention score to values of 0 or 1, which is a differentiable function and can be incorporated into the back-propagation of the learning process.

Finally, we combine the Co-Attention ranking loss Eq. (Equation 4.3) and the contrastive loss Eq. (Equation 4.6) as the training objective of CCANet:

$$L = L_{Rank} + L_C.$$  

(4.8)

4.4 Experiment Results

4.4.1 Movie Key Moment Detection Results

Dataset and Evaluation Metric. We evaluate our CCANet on the constructed dataset TMDD. Under a specific movie genre containing 50 movies, we randomly split the movies into the training and test set, containing 45 and 5 movies respectively. In the experiment, we repeat the split three times and report the average across three runs as the final result. During test, the movie shots are ranked based on the predicted score and then compared with the human-verified “key moment” ground-truth obtained by matching shots between trailers and movies as described in Sec. Subsection 4.3.1.

For the evaluation metric, we calculate Average Precision (AP) on each test video to measure the shot ranking performance. In order to get a fine-grain local view on the ranking performance on each video, we adapt AP to a $\text{Rank}@N$ metric which can be illustrated in Figure 4.4(Left). As seen, we examine the ranking AP within every $N$ consecutive
Figure 4.4: **Left:** Rank@N. We calculate AP within every $N$ consecutive shots in a full-length movie and average them as the overall performance metric, offering a local-view on the ranking performance. The top row lists trailer (blue) and non-trailer (grey) shots in a movie along the timeline before ranking. The middle and bottom illustrate the ideal Rank@N results. **Right:** the “hard” annotation brings about ambiguity in the labels. A trailer shot and its four visually similar movie shots are shown. The movie shot marked by the green border is labeled as positive and the rest shots are negative.

shots in the movie and average them across the entire movie as the performance metric. Rank@Global is equivalent to AP where $N$ equals to the number of shots in the movie. We calculate the results on each movie and average them across all test movies as the overall performance.

**Feature Extraction.** The 3D CNN [144] (S3D) with a ResNet-34 [2] backbone pre-trained on Kinetics-400 dataset [113] are used to compute the input features. We use the output after the global pooling of the final convolution layer and a shot is represented by a feature of 512 dimensions, same as the work [140]. Specifically, a feature vector is extracted from a snippet covering 16 consecutive frames. The snippet features are averaged to represent the shot, where a snippet belongs to specific shot if $>70\%$ frames of the snippet are covered by the shot.

**Implementation Details.** We implement our model with PyTorch\(^2\), and optimize the loss with Adam optimizer [145] for 50 epochs. We use a batch size of 2048 and set the base learning rate to 0.001. With a single NVIDIA K80 gpu, the total feature extraction time for a 4-second shot is 0.05s. After extracting features, the time to train a ranking model for Drama movies is one hour, which contains $480K$ snippets in a total duration of $\sim100$ hours. At test time, it takes 0.04s to score a batch of snippets after feature extraction.

\(^2\)https://pytorch.org/
Table 4.1: The trailer moment detection results on TMDD. “Sup”, “PL” and “CoA” denote the different approaches, including fully-supervised, Pseudo-Label and our Co-Attention, with the basic 3D features [144]. The “Sup+CA”, “PL+CA” and “CCANet” denote that the shot features in ‘Sup’, “PL” and “CoA” are augmented with our proposed Contrastive Attention module. The terms “Act”, “Dra” and “ScF” refer to the movie categories, i.e., Action, Drama and Sci-Fi, and “Avg” indicates the “Average” result across categories. The subscripts “10”, “20” and “GL” indicate different evaluation metrics of Rank@10, Rank@20 and Rank@Global.

Comparison Baselines. We compare our CCANet to two baselines, where the experiment settings such as learning rate, batch size and so on, follow the same practice as CCANet.

- Fully Supervised MTMD. We assume the annotated trailer shots are accessible. Then we can perform supervised training as the VHD approaches described in Sec Section 4.1. The movie shots annotated as trailer moment are the positive samples. For each positive sample, we sample 20 negative (non-trailer) shots, forming a set of pairs to train the ranking model as in Eq. (Equation 4.1).

- Weakly Supervised MTMD with Pseudo Label. We also compare CCANet to a weakly supervised approach using the Pseudo Label, which does not require access to manual annotations. We offline calculate the visual similarity between trailer and movie shots. The movie shots having the high similarity to the trailer are regarded as the positive samples, and those with low similarity as the negatives.

Results. Table Table 4.1 presents the trailer moment detection results of different approaches. As seen, by using our Co-Attention (CoA) module alone, our approach substantially outperforms the two baselines. Notably, CoA achieves \( \sim 6\% \) Rank@Global margin over the supervised approach. The trend is that the Rank@N drops as \( N \) increases, and Rank@Global is the lowest compared to \( N=10, 20 \). The performance drop is attributed
Figure 4.5: Performance variance with respect to $\lambda$. We change the $\lambda$ value and report the performance of the proposed CCANet. The evaluation metric is Rank@10.

<table>
<thead>
<tr>
<th></th>
<th>Act$_{GL}$</th>
<th>Dra$_{GL}$</th>
<th>ScF$_{GL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoA</td>
<td>0.228</td>
<td>0.221</td>
<td>0.208</td>
</tr>
<tr>
<td>$+$FeaAug</td>
<td>0.255</td>
<td>0.230</td>
<td>0.236</td>
</tr>
<tr>
<td>$+$FeaAug+L$_C$</td>
<td><strong>0.271</strong></td>
<td><strong>0.246</strong></td>
<td><strong>0.242</strong></td>
</tr>
</tbody>
</table>

Table 4.2: Rank@$Global$ of the proposed CoA approach with different feature encoding strategies.

to the fact that increasing $N$ involves more negative samples for ranking. Especially, the fully-supervised approach drops the most at the global ranking metric. An explanation is that it suffers from the “hard” annotations provided by annotators. The “hard” means that a movie shot is considered as a positive sample only when it is an exact trailer moment. As shown in Figure 4.4 (Right), only the shot at the top is annotated as the trailer shot (positive sample) as it is an exact match to the trailer while the other three are regarded as negative samples. Forcing those movie shots to be separable largely in the feature space brings the ambiguity to train the model. Our CoA module tackles this problem by assigning the soft labels to the data and a training pair with the closer attention scores contributes less to the loss calculation.

We also apply the proposed Contrastive Attention module to augmenting the features in all comparison approaches. In Table Table 4.1, the models with augmented features show superior performance over their origins with the 3D features only [144]. The results validate that exploring the relations among different shots can enhance the feature representation and boost the performance.

**Impact of parameter $\lambda$.** In Eq. (Equation 4.3), we introduce a heuristic parameter $\lambda$ to weight the validness of a training pair. The impact of $\lambda$ to CCANet’s performance is shown in Figure 4.5, where we report the results measured by Rank@10 and choose the value
leading to the best performance. As can be seen, the performance is not sensitive to the value variation of $\lambda$ and we set the value of $\lambda = 1.5$ as default.

**Ablation study.** In Table Figure 4.2, we perform ablation study to examine our key contribution of Contrastive Attention by evaluating the CoA approach with three variants of shot feature encoding: 1) CoA uses the 3D feature only [144]; 2) CoA+$Fea Aug$ augments features as Eq (Equation 4.5) without contrastive loss $L_C$; 3) CoA+$Fea Aug+L_C$ is our CCANet. The $\sim 2\%$ performance gain from $+Fea Aug$ over CoA shows the importance of exploring the relations among clips for feature encoding. Further, our CCANet consistently improves CoA+$Fea Aug$. Our interpretation is that the loss $L_C$ is introduced to guide the attention to be contrastive, encouraging the features of trailer shots to form a compact clique in the feature space and more distinguishable from the features of the non-trailer shots. As a result, it relieves the difficulty of learning the rank model and make CCANet achieve the best performance.

### 4.4.2 Video Highlight Detection Results

We also evaluate the proposed Contrastive Attention$^3$ on the VHD benchmarks, demonstrating its effectiveness. VHD has a similar goal to MTMD, aiming to detect the highlight moments in video which are supposed to be noticeable among the non-highlight moments, which naturally manifest the contrastive relations. We follow the work [140] to choose two challenging public video highlight detection datasets including YouTube Highlights [118] and TVSum [119]. The trained highlight detectors are domain-specific [140].

**Datasets.** YouTube Highlights dataset [118] contains six domain-specific categories: surfing, skating, skiing, gymnastics, parkour, and dog. Each domain consists of $\sim 100$ videos and the total duration is $\sim 1430$ minutes. Each video is divided into multiple clips and humans annotate whether a clip contains a specific category. TVSum [119] is collected from YouTube using 10 queries and consists of 50 videos in total from domains such as changing

---

$^3$Our Co-Attention module is not applicable for the VHD task since there are no video pairs in VHD as the trailer-movie pairs in MTMD.
vehicle tire, grooming an animal, making sandwiches, parade, etc. (see Table 4.4). We follow the works [140, 137] to average the frame-level scores to obtain the shot-level scores, and then select the top 50% shots from each video to build the ground-truth. Finally, the highlights selected by our approach are compared with the ground-truth.

**Evaluation Metric and Baselines.** We follow the works in [140, 118], using the mean Average Precision (mAP) and mAP at top-5 to evaluate the highlight detection results on Youtube Highlights [118] and TVSum [119], respectively. We compare with eleven state-of-the-art approaches, which are categorized into unsupervised and supervised approaches. Those previous works’ results are reported by the original papers. Specifically, We compare with the unsupervised approaches of RRAE [139], MBF [146], CVS [147], SG [148], DeSumNet(DSN) [137], VESD [149] and LM [140]. In particular, the latest approach LM [140] uses the duration signal as the supervision to train a ranking model and training data contains around 10M Instagram videos.

We also include the supervised approaches, e.g. KVS [138], seqDPP [150], Sub-Mod [151], CLA, GIFs [135] and LSVM [118]. The latent SVM (LSVM) [118] has the same supervised ranking loss as ours, but LSVM uses the classic visual features while our features are augmented by the Contrastive Attention module.

**Results on Youtube Highlights.** Table 4.3 shows the results on YouTube Highlights dataset [118]. All the baseline results are quoted from the original papers. Our approach achieves the best performance and substantially improves those baselines with a large margin. Notably, our approach outperforms the following best performing LM [140] by 12.7% and CLA by 27.5%, with relative gains of 23% and 66%, where both LM and CLA models are trained on the additional 10M Instagram videos. We also achieve 15.5% performance gain over the LSVM. The LSVM [118] trains a ranking model with domain-specific manually annotated data, but its basic visual feature is limited to capture the feature distribution. Our proposed Contrastive Attention module explicitly models the relations between highlights and non-highlights so that highlight feature can form a compact clique in the feature space.
Table 4.3: The highlight detection mAP on YouTube Highlight dataset. Avg. is the average mAP over all the domains. Our approach outperforms all the baselines.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>0.49</td>
<td>0.308</td>
<td>0.6</td>
<td>0.502</td>
<td>0.579</td>
<td><strong>0.633</strong></td>
</tr>
<tr>
<td>gymnastic</td>
<td>0.35</td>
<td>0.335</td>
<td>0.41</td>
<td>0.217</td>
<td>0.417</td>
<td><strong>0.825</strong></td>
</tr>
<tr>
<td>parkour</td>
<td>0.5</td>
<td>0.54</td>
<td>0.61</td>
<td>0.309</td>
<td><strong>0.67</strong></td>
<td>0.623</td>
</tr>
<tr>
<td>skating</td>
<td>0.25</td>
<td>0.554</td>
<td><strong>0.62</strong></td>
<td>0.505</td>
<td>0.578</td>
<td>0.529</td>
</tr>
<tr>
<td>skiing</td>
<td>0.22</td>
<td>0.328</td>
<td>0.36</td>
<td>0.379</td>
<td>0.486</td>
<td><strong>0.745</strong></td>
</tr>
<tr>
<td>surfing</td>
<td>0.49</td>
<td>0.541</td>
<td>0.61</td>
<td>0.584</td>
<td>0.651</td>
<td><strong>0.793</strong></td>
</tr>
<tr>
<td>Avg.</td>
<td><strong>0.383</strong></td>
<td><strong>0.464</strong></td>
<td><strong>0.536</strong></td>
<td><strong>0.416</strong></td>
<td><strong>0.564</strong></td>
<td><strong>0.691</strong></td>
</tr>
</tbody>
</table>

Table 4.4: The highlight detection top-5 mAP score on TVSum [119]. The '-' means that mAP value is not provided in the original paper.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle tire</td>
<td>0.295</td>
<td>0.353</td>
<td>0.328</td>
<td>0.423</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.294</td>
<td>0.559</td>
<td>0.613</td>
<td></td>
</tr>
<tr>
<td>Vehicle unstuck</td>
<td>0.357</td>
<td>0.441</td>
<td>0.413</td>
<td>0.472</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.246</td>
<td>0.429</td>
<td>0.546</td>
<td></td>
</tr>
<tr>
<td>Grooming animal</td>
<td>0.325</td>
<td>0.402</td>
<td>0.379</td>
<td>0.475</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.590</td>
<td>0.612</td>
<td>0.657</td>
<td></td>
</tr>
<tr>
<td>Making sandwich</td>
<td>0.412</td>
<td>0.417</td>
<td>0.398</td>
<td>0.489</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.433</td>
<td>0.540</td>
<td>0.608</td>
<td></td>
</tr>
<tr>
<td>Parkour</td>
<td>0.318</td>
<td>0.382</td>
<td>0.354</td>
<td>0.456</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.505</td>
<td><strong>0.604</strong></td>
<td>0.591</td>
<td></td>
</tr>
<tr>
<td>Parade</td>
<td>0.334</td>
<td>0.403</td>
<td>0.381</td>
<td>0.473</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.491</td>
<td>0.475</td>
<td><strong>0.701</strong></td>
<td></td>
</tr>
<tr>
<td>Flash mob</td>
<td>0.365</td>
<td>0.397</td>
<td>0.365</td>
<td>0.464</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.430</td>
<td>0.432</td>
<td><strong>0.582</strong></td>
<td></td>
</tr>
<tr>
<td>Beekeeping</td>
<td>0.313</td>
<td>0.342</td>
<td>0.326</td>
<td>0.417</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.517</td>
<td><strong>0.663</strong></td>
<td>0.647</td>
<td></td>
</tr>
<tr>
<td>Bike tricks</td>
<td>0.365</td>
<td>0.419</td>
<td>0.402</td>
<td>0.483</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.578</td>
<td>0.691</td>
<td><strong>0.656</strong></td>
<td></td>
</tr>
<tr>
<td>Dog show</td>
<td>0.357</td>
<td>0.394</td>
<td>0.378</td>
<td>0.466</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.382</td>
<td>0.626</td>
<td><strong>0.681</strong></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td><strong>0.345</strong></td>
<td><strong>0.398</strong></td>
<td><strong>0.372</strong></td>
<td><strong>0.462</strong></td>
<td><strong>0.424</strong></td>
<td><strong>0.423</strong></td>
<td><strong>0.447</strong></td>
<td><strong>0.461</strong></td>
<td><strong>0.447</strong></td>
<td><strong>0.563</strong></td>
<td><strong>0.628</strong></td>
</tr>
</tbody>
</table>

Results on TVSum. Table 4.4 shows the results on TVSum dataset [119]. Our approach outperforms all the baselines by a noticeable margin. In particular, our results achieve 6.5% mAP higher than the following best performing approach LM [140]. Regarding the supervised approaches, we also outperform SubMod [151] by 16.7%, where the SubMod [151] proposes an adapted submodular function with structured learning for the highlight detection.

4.4.3 Understanding the Co-Contrastive Attention

Co-Attention between trailer and movie shots. We examine the Co-Attention scores between the trailer and movie shots. In Figure 4.6, the score achieves the highest when the
trailer moments exactly comes from the movie shots. Our model assigns reasonable high scores to the shots which are visually similar to the trailer moment.

**Feature augmented by Contrastive Attention.** In Figure 4.7, we plot the UMAP embedding [152] of the basic 3D features and the augmented features with the Contrastive Attention on domains of “Surfing” and “Gymnastics” from Youtube Highlights [118] dataset. As can be seen, the augmented highlight and non-highlight features are more separable in the feature space, which eases the difficulty of learning a robust model for highlight detection, resulting in the performance improvement in both domains.

### 4.5 Conclusion

In this work, we propose the CCANet to address the problem of learning the trailer moments from movies. Our approach utilizes Co-Attention scores as supervision, which does not require expensive human-annotations. Additionally, we introduce the Contrastive Attention
module to augment the video features, equipping the model with the capacity of capturing the contrastive relation between the trailer and non-trailer moments. To evaluate our approach, we are the first to collect the dataset, TMDD. The effectiveness of our approach is demonstrated by the performance gain not only on our collected data but also on the public benchmarks. The results on TMDD also demonstrate there is a large room for improvements in trailer moment detection, e.g. multi-modality might be used to boost the robustness, which is part of our future work.
5.1 Introduction

Research shows that when humans communicate, more social meaning comes from non-verbal than verbal cues. Among the nonverbal modalities, the face is the one upon which people typically rely [153]. Facial expressions convey one’s identity, display emotions, show status, give context, open or shut down a conversation, signal approval, and reveal the strength of conviction, among other things. People rely on facial cues to glean both intentional and unintentional meaning. With so much communicated by the face, it is natural that facial expressions have been investigated for possible cues to deception for decades [154, 155, 156]. With advances in computer vision has come the possibility of detecting facial movement variations on a more granular scale than the human eye can perceive, and with it, the discovery of deception indicators not normally directly detectable by human perception [157, 158]. Although much deception research has focused on the emotional potential of the face, searching for micro-level “leaked” indicators that betray concealed true emotions [159], the face can reveal far more signals related to deception. It may reveal signs of cognitive effort and efforts to retrieve information from memory as a speaker attempts to formulate a believable verbal statement [160]. For example, blink patterns and lip presses may be associated with a speaker’s thought processes. The face may signal not just internal emotional states such as fear or distress, but also affect directed toward another such as contempt or dislike. Nose flares and inauthentic smiles may signify these states. Communicators may also signal their attentiveness to others or their desire for a speaking turn [161]. Because people are aware that others’ gaze is directed to the face, deceivers
try to control their face and may, in the process, inadvertently over control it, producing a pattern of rigidity [162, 163]. Head movement, facial animation and gaze patterns may all reflect this “freezing” of activity. However, if deceivers have opportunities to rehearse, plan or mentally edit what they say, any temporary missteps may be repaired [164]. Given the fluidity of facial expressions, temporal patterns can also be telling. For instance, blink patterns vary during versus after lying, and the onset and offset of smiles may differ by truth-tellers versus liars.

In previous research, to automatically decipher the meaning in nonverbal human communication using computer vision methods, researchers first applied models inspired by communication theory. However, the computer vision-based analysis’s underlying human-defined features were incomplete due to non-linearity and the multi-scale nature of the problem. The recent use of neural nets has addressed discovering the features associated with the computer vision-based analysis of nonverbal communication. It has improved the recognition of desired events significantly during nonverbal communication, such as truth-telling.

5.2 Related Work

Visualizing CNNs. A number of previous works have been proposed to visualize the internal representations offline in an attempt to better understand the model. In [28] [165] and [30], they compute the gradient of the prediction w.r.t the specific CNN unit, i.e. the input image, to highlight the important pixels. Specifically, Simonyan et al. [28] visualize partial derivatives of predicted class scores w.r.t. pixel intensities, while Guided Backpropagation [165] and Deconvolution [30] make modifications to ‘raw’ gradients that result in the better visualization. Despite producing fine-grained visualizations, these methods are not class-discriminative.

Erhan et al. [25] synthesize the images to maximally activate a network unit and Mahendran et al. [26], Dosovitskiy et al. [27] analyze the visual coding so as to invert latent
representation. Although these can be high-resolution and class-discriminative, they visualize a model overall and not predictions for specific input images.

Our work is mainly inspired by recent works [14, 9, 15] addressing the class-discriminative attention maps. CAM [14] generates the class activation maps highlighting the task-relevant region by replacing fully-connected layers with convolution and global average pooling. A drawback of CAM is the low flexibility, which requires retraining of the classifiers and feature maps to directly precede softmax layers. Hence it is unable to be applicable to any feature layers. Grad-CAM [9] is proposed to address this issue. Without retraining and changing network architecture, Grad-CAM generates the class activation maps by a weighted combination of the feature maps in different channels. The weights are computed by the averaging of the gradient of the final prediction w.r.t the pixels in the feature map. According to our observation, simple averaging is unable to measure the channel importance properly, which causes a large attention inconsistency among different feature layers. Grad-CAM++ [15] proposed a better class activation map by modifying the weight computation while its high computation cost of calculating the second and third derivatives makes it hard to be used to train the model.

**Video Highlight Detection** is highly related to our research topic since we intend to extract a brief synopsis containing segments of special interest from a video [134]. Many earlier approaches have primarily been focused on highlighting sports videos. A latent SVM model is employed to detect highlights by learning from pairs of raw and edited videos [118]. Success of deep learning also imparted improved performance in highlight detection [139]. However, most of these techniques may not generalize well to web videos since they are either based on heuristic rules or require huge amount of human-crafted labelling data which are difficult to collect in many cases. In our *The Resistance* games, we only have video-level annotations of players’ roles (Spy/Villager) without knowledge about when and where, in the untrimmed videos, players show the notable facial movements for the roles. Finding those movements are important for understanding human behaviours during communication.
Figure 5.1: The attention maps are generated via weighted combination of the feature maps at the specific layers. The weights measure the importance of the features, computed according to the gradients, where we take derivative of the class score w.r.t the feature maps.

To achieve this goal, we incorporate the interpretation in the learning to discover those pixels and related frames which are discriminative and contributed the most to the NN’s prediction for the players roles.

5.3 Methodology

In this section, we describe the detail of how to extract the class-discriminative attention map for the videos. The procedure is illustrated by Figure 5.1. Motivated by the work of Grad-CAM [9] and Grad-CAM++ [15], we use the gradient to measure the importance of each feature map pixel to having the model classify the input image as class $c$. For the gradient of the class score $Y^c$ is computed by taking derivative w.r.t feature map $F^k$ in $k$-th channels, i.e. $(\partial Y^c)/\partial F^k$). The pixel importance is denoted as $(\partial Y^c)/(\partial F^k_{ij})$. In [9, 15], the gradients are used to compute the channel-wise weights for combining the feature maps from different channels, generating the attention map of the last feature layer $A_{Grad-CAM}$:

$$A_{Grad-CAM} = ReLu\left(\sum_k \alpha_k^c F^k\right),$$  \hspace{1cm} (5.1)
where $\alpha_{k}^{c}$ indicates the importance of the feature map $F^{k}$ in the $k$-th channel. In [9], the weight $\alpha_{k}^{c}$ is a global average of pixel importance in the feature map:

\[
\alpha_{k}^{c} = \frac{1}{Z} \sum_{i} \sum_{j} \frac{\partial Y^{c}}{\partial F_{ij}^{k}}
\]

(5.2)

where $Z$ indicates the total number of pixels in feature map $F^{k}$. In [15], higher order derivatives (second and third) involved to compute the channel weights increase the computational costs.

Besides only generating the attention map of the last feature layer as in [9, 15], we compute the category-oriented attention map for the intermediate layers. In terms of the interpretability, we propose two attention mechanisms for any feature layer with low computational cost, modeling the channel and pixel-wise attention respectively. Then, we combine the model’s channel and pixel-wise attention to generate the final response map for the input video.

**Channel-wise attention** Different from Equation 5.1 that the Grad-CAM uses the gradients of all the pixels to compute the channel weight, we only select the positive gradients and average them to obtain the channel-wise importance:

\[
\alpha_{k}^{c} = \frac{1}{Z} \sum_{i} \sum_{j} \text{ReLU}\left( \frac{\partial Y^{c}}{\partial F_{ij}^{k}} \right)
\]

(5.3)

The intuition is that the positive gradients model the pixels where the intensity increasing has positive impact on the final prediction score [15]. Substitute Equation 5.3 to Equation 5.1, we have the channel-wise class-discriminative attention $A_{ch}$.

\[
A_{ch} = \frac{1}{Z} \text{ReLU}\left( \sum_{k} \sum_{i} \sum_{j} \text{ReLU}\left( \frac{\partial Y^{c}}{\partial F_{ij}^{k}} \right) F^{k} \right)
\]

(5.4)

**Pixel-wise attention** Attention proposed by the previous works [14, 9, 15] is computed in the channel-wise way, where the pixels within the same channel share the same weight.
for feature maps combination. Besides channel-wise attention, we also find that pixel-wise attention demonstrates better guidance when training a model in low-quality images. Specifically, each channel acts as an expert to vote the pixel importance in the attention map. In the feature map $F^k$, the pixel intensity is scaled by its importance measured as $(\partial Y^c)/(\partial F^k_{ij})$ and the averaging is performed across channels to obtain the pixel-wise attention:

$$A_{px} = ReLu\left(\frac{1}{K} \sum_k < \frac{\partial Y_c}{\partial F^k}, F^k > \right),$$

(5.5)

where the $< \frac{\partial Y_c}{\partial F^k}, F^k >$ indicates the element-wise multiplication between the gradient and feature maps.

The harmonic attention According to our observation, the pixel-wise attention captures more high-frequency items, and the channel-wise attention maps are smoother. Those two types of attention are complimentary where the $A_{px}$ highlight the important pixels which are ignored by $A_{ch}$ due to the low value averaged channel weights. Hence, we propose to combine the $A_{px}$ and $A_{ch}$, generating the harmonic attention $A$. Empirically, applying the pixel-wise weighting first and then computing the channel-wise attention as Eq. Equation 5.3 achieves better performance. The proposed harmonic attention is formulated as:

$$A = \frac{1}{Z} ReLU\left(\sum_k \sum_i \sum_j ReLU\left(\frac{\partial Y_c}{\partial F^k_{ij}} \right) < \frac{\partial Y_c}{\partial F^k}, F^k > \right)$$

(5.6)

5.3.1 Training a 3D convolutional Neural Network for Spy Detection

We formulate spy detection as a binary classification problem. Given a video sequence, we apply a 3D Convolutional Neural Network (C3D) [166] to classify the player as a spy or villager. Specifically, we crop the players’ faces, and the C3D takes a facial video clip as input, predicting the probability for his/her role. The cropped face frames are normalized into the size of $112 \times 112$. In the C3D architecture, we design the model having 8 convolutions, 5 max-pooling, and 2 fully connected layers followed by a softmax layer. The 3D convolution
Figure 5.2: The details of C3D architectures

kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. The number of filters is denoted in each box, as shown in Figure 5.2. The 3D pooling layers are denoted from pool1 to pool5. All pooling kernels are $2 \times 2 \times 2$, except for pool1 is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units. The output has 2 dimensions for binary classification. The model training and testing are conducted by using PyTorch and NVIDIA K80 XGPUs.

5.4 Experiments

5.4.1 Dataset

We tested our method on data from a real-world game, where the goal is to examine deceivers’ strategies and truth-tellers deception detection abilities. Groups of participants were sought to play a board game adapted from Resistance, during which players in the roles of Villagers (truth-tellers) and Spies (deceivers) competed to win missions. The games were played in eight different locales across the world to detect cultural differences in communication strategies and patterns. In the following, we present details on dataset collection in terms of different locations, participants, procedure, game play and measurements.

Participants

There are 693 participants recruited via email, message boards and advertisements from public universities in the Southwestern US (9 games; n = 59), Western US (11 games; n = 67), Northeastern US (10 games; n = 74), Israel (10 games; n = 71), Singapore (12 games; n = 84), Fiji (14 games, n = 106), Hong Kong (15 games, n = 115), and Zambia (15 games, n = 117). The sample was 59% female and was ethnically diverse, with the biggest groups being Asian (38%) or white, non-Hispanic (18%). Nationalities represented 41 different
countries. Participants were required to be proficient English speakers. Each game was approximately 2 hours long.

**Procedure**

Participants enrolled using an online scheduling system. Groups ranged from five to eight participants. Before arrival at the site, participants completed consent forms, cultural measures, and demographic questions. Upon arrival, participants were randomly assigned to one of eight computers equipped with a desk, a computer tablet with a built-in webcam, and a chair. Participants were informed that they would be filmed by the cameras.

Each group had a facilitator who explained the rules of the game. The interaction began with an ice-breaker activity, after which players rated each other on scales meant to capture baseline perceptions of dominance, composure, and trustworthiness. Participants took part in the game for an hour, during which they played between three and eight rounds. After the second, fourth, and sixth rounds, and at the end of the game, participants again completed ratings of one another and identified who they thought were the spies. Participants were paid for participating and received additional financial incentives for performing well.

**Game play**

Similar to [167], we adapted a version of the Mafia game that closely resembles the board game The Resistance. We pilot tested several versions of the game to ensure the game best met the needs of the research questions. Players were randomly and secretly assigned to play deceivers (called "Spies"), or truth-tellers (called "Villagers"). In games of five or six players, two were assigned to be Spies, and in games of seven or eight players, three were assigned to be Spies. The goal of Villagers was to remove Spies from their community; the goal of Spies was to undermine the missions of the Villagers. Spies were aware of who the other Spies were, but Villagers did not. Villagers had to depend on shared information to deduce the other players’ identities within the game.

Players completed a series of "missions" by forming teams of varying size. At the beginning of each round, players elected a leader, who then chose other players for these
missions based on who they thought would help them win the game. All players voted to approve or reject the leader, then voted on the leader’s proposed team. Players voted secretly on their computer and publicly by a show of hands. Facilitators would announce if there was a discrepancy in public and private votes, thus informing participants when deception occurred. The leader chooses players to go on a mission secretly and vote for the mission to succeed or fail. Villagers won rounds by figuring out who the spies were and excluding them from the mission teams. Spies won rounds by causing mission failures. The ultimate winner (Spies or Villagers) was determined by which team won the most rounds. Additionally, players won monetary rewards by being voted as a leader or winning the game.

Measures

We design several measurements for monitoring the game play, including Game Outcome, Trust, Dominance and Previous Game Experience.

*Game Outcome:* In [167] Mafia study, they regard the deception detection success as the truth-tellers winning the game (i.e., if the truth-tellers win, they must have accurately detected deception). Similarly, in this study, game outcome was a dichotomous variable measuring whether or not Spies or Villagers won the game.

*Trust:* The extent to which participants trusted each of the other players was measured using a single-item repeated measure, which was asked after the ice-breaker, and then every even-numbered round during the game. The item read: Please rate how much you trust each player. Are they trustworthy or suspicious? A rating of 5 would mean they seem honest, reliable and truthful and 1 would mean you thought they were dishonest, unreliable and deceitful (1 Not at all to 5 Very much; Mean = 3.29, SD = 1.36). Because participants responded to this item three to five times about each of the other players, we chose to use a single item in order to avoid fatigue.

*Dominance:* The extent to which participants found other players to be dominant was measured using a single-item repeated measure (after the icebreaker and each of the even numbered rounds). Participants read the following text: Please rate how dominant each
player is. Are they active and forceful or passive and quiet? A rating of 5 would mean you thought they were assertive, active, talkative, and persuasive. A score of 1 would mean you thought they were unassertive, passive, quiet and not influential. We got the statistical of Dominance as Mean = 3.28 and SD = .87.

Previous Game Experience: Participants’ previous experience playing similar games was evaluated after the completion of the post-game measures. Participants indicated that they had or had not played a similar game. In this study, 54.2% said they had not played a similar game before.

5.4.2 Results of Spy Detection

In the experiments, 280 players’ videos are collected for Spy/Villager prediction, including 110 spies and 170 villagers, where the players with different cultural backgrounds are mixed. We segment the video clip of the first game round for training and testing. The total length is 84000 seconds (1400min).

We randomly select the 10%/20% videos as testing data and the rest as the training, where no duplicate players appear in both training and validation set. For each setting, the experiments are conducted 5-times cross-validation and the results are reported in Table 5.1 and Table 5.2. Those two tables show the Spy/Villager prediction accuracy with two different frame sampling mechanisms, random sampling, and attention-guided frame sampling.

Random Sampling During training, given a video file, we random sample 16 frames as the input of C3D, and each frame is resized to 112x112. The temporal order is kept among the selected frames. The prediction accuracy is shown in Table 5.1, where the C3D model performs better than the random guess of 60% (170/280), in the margin of \( \sim 3\% \) and \( \sim 5\% \).

Attention-guided Sampling The crucial difference between our Spy/Village prediction and the use of the conventional image or video classification is that even with the label (spy or villager), a human has a hard time explaining why the players are classified as ’Spies’. In most cases, spies and villagers have very similar behaviors, which means the data is not
discriminative. As the high accuracy of spy prediction is one of our goals, finding where and when the players show the visual cues for ‘being a spy’ is our work’s goal. As in Table 5.1, the trained deep neural net model demonstrates better performance than a random guess, which motivates us to interpret the model so as to understand what visual patterns make the model predict the players as ‘Spies’. Given the trained C3D model, we apply the proposed harmonic attention mechanism to compute the frame importance via averaging the attention maps. Instead of random sampling the frames in equal probabilities, we sample the frames according to the importance, leading to higher chances to sample the frames with more contribution to Spy/Villager prediction.

We keep all the parameters the same for training models with the two different frame sampling approaches. Table 5.1 shows our classification results when we randomly selected the video clips from the data. Table 5.2 shows the classification results when we retrain the model based on our attention discovered video frames. The results show clearly that the model trained with attention-guided frame sampling outperforms the one with random sampling in the notable margins, \( \sim 2\% \) and \( \sim 4\% \) in testing/training splitting of 1:9 and 2:8 respectively. The performance-boosting validates our attention mechanism’s effectiveness to identify the potential frames where spies and villagers show notable discriminative visual

<table>
<thead>
<tr>
<th>#Validation/Training Games</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>65.43(± 0.27)</td>
</tr>
<tr>
<td>2/8</td>
<td>62.28(± 0.30)</td>
</tr>
</tbody>
</table>

Table 5.1: The Spy/Villager prediction accuracy reported on the two different dataset splitting. The training data is randomly sampled without attention knowledge.

<table>
<thead>
<tr>
<th>#Validation/Training Games</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>67.85(± 0.25)</td>
</tr>
<tr>
<td>2/8</td>
<td>67.03(± 0.28)</td>
</tr>
</tbody>
</table>

Table 5.2: The Spy/Villager prediction accuracy reported on the two different dataset splitting. The model is trained with the video frames selected according to the model attention.
Figure 5.3: We compare what the model attends to known Action Units which are useful in communication research regarding face and head. The attended facial cues are coded as facial action units (AU). All examples are from spies.

signals so that it is easier to train a model with better accuracy. Besides the quantitative results, we also apply the attention map to identify important pixels and visualize them in the next subsection.

5.4.3 Attention and Deception Cues

In Figure 5.3 we show promising qualitative results because our attention NN can discover cues related to what is known from communication theory for deception. In Figure 5.3 we show some Facial Action Units related to spies extracted from the discovered frames and the respective probabilities, i.e., AUs:13,20,24,45. The players showing such AUs are more likely to be classified as Spies. According to the communication theory, AUs 20 and 45 are related to deception, which is consistent with our expectation that spies are more willing to lie, but not always. In Figure 5.4 we show the ability of our network to attend to different cues for a spy and a villager, which are also consistent with the current communication theory [156, 157] on deception. Figure 5.4 clearly shows cues and respective pixel probabilities
Figure 5.4: What the model attends to for "Spy" vs. "Villager". We show the comparison of attention maps between spy and villager. The model can attend to small facial movements related to deception, like eye blinking in the bottom left row (spies). And at the top row (spies and villagers), the model detects fake and real smiles to classify the two types of players’ roles correctly.

related to deception, such as eyes closed, fake smiles, changes in lips. In particular, we show the comparison of model attention between spies and villagers. For example, our approach can attend to small facial movements related to deception, like eye blinking in the bottom left case (spies). At the top row (spies and villagers), the model detects the fake and real smiles to correctly classify the two types of players’ roles. These initial encouraging results show that we can extract cues and AUs related to communication and deception theory without using prior known cues. They provide cues that are human-interpretable and can be used in many other types of applications.

5.5 Conclusion

In this paper, we presented a novel attention-based neural network (NN) that discovers through learning in a video sequence the most discriminative frames and related pixel probabilities and AUs that contributed the most to the final class inference of the neural net. We applied our method to facial videos of a variant of the Resistance game collected in various countries where the players assume the roles of deceivers (spies) vs. truth-tellers (villagers). We demonstrated for the first time that it is possible to discover the
frames and AUs that contributed the most to the NN’s class decision on several hours of video testing. The results are consistent with the current communication theory on nonverbal communication. They can be used in future studies to discover static and dynamic relationships among cues and AUs currently unknown.
CHAPTER 6
CONCLUSION

In this dissertation, we address the challenges of CNN-based visual learning in which attention mechanisms can be used as crucial approaches to deal with those problems. We specifically focus on three aspects: 1) Generate visual explanation to interpret CNN model and reason human facial behaviors; 2) Incorporate attention mechanisms into the learning process, guiding model training and bridging multiple vision tasks; 3) Applying attention as weak supervision to reduce annotation budget.

• We propose channel-weighted attention $A_{ch}$, which has better localizability and avoids higher-order derivatives computation, compared to existing approaches for attention-driven learning. And we apply our method to facial videos of a variant of the Resistance game, demonstrating that it is possible to discover the frames and AUs that contributed the most to the NN’s class decision on several hours of video testing. The results are consistent with the current communication theory on nonverbal communication. They can be used in future studies to discover static and dynamic relationships among cues and AUs currently unknown.

• We propose a new framework, ICASC, which makes class-discriminative attention a principled part of training a CNN model for image classification. Our proposed attention separation loss and attention consistency loss provide supervisory signals during training, resulting in improved model discriminability and reduced visual confusion.

• We address the possibility of taking advantage of the attention to bridge different vision tasks into a unified learning process. Specifically, we propose a coupled encoder-decoder neural network to detect faces and localize landmarks jointly. The
performance on both of the two tasks is very competitive while sometimes better than some state-of-the-art methods.

- We propose the CCANet to address the problem of learning the trailer moments from movies. Our approach utilizes Co-Attention scores as supervision, which does not require expensive human-annotations. Additionally, we introduce the Contrastive Attention module to augment the video features, equipping the model to capture the contrastive relation between the trailer and non-trailer moments.

Our dissertation work was an attempt to unitize attention to deal with the challenges of visual learning. The encouraging potential future work might contain several directions. In Chapter 2, while we select the last two feature layers, which contain the most semantic information to generate the attention maps, ICASC is flexible w.r.t. layer choices for attention generation, and we plan to study the impact of various layer choices in the future. In Chapter 3, future work might focus on how to formulate the two tasks as a single optimization problem. In Chapter 4, the results on TMDD demonstrate there is a large room for improvements in trailer moment detection, e.g., multi-modality might be used to boost the robustness, which is part of our future work. Also, applying attention mechanism to cross-domain data, e.g., text, audio, and image, resulting in model interpretation closed human perception might be another interesting research direction.
REFERENCES


[68] B. Yang, J. Yan, Z. Lei, and S. Z. L. C. C. Features, “Ieee international conference on computer vision (iccv),” vol. 2015,


