5-2023

Improving Classification in Single and Multi-View Images

Hadi Kanaan Hadi Salman
University of Arkansas, Fayetteville

Improving Classification in Single and Multi-View Images

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering

by

Hadi Kanaan Hadi Salman
Cairo University
Bachelor of Engineering in Computer and Electrical Engineering,
2014 University of Arkansas, Little Rock
Master of Science in Systems Engineering, 2018

May 2023
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

John Gauch, Ph.D.
Dissertation Director

Susan Gauch, Ph.D.
Committee Member

Thi Hoang Ngan Le, Ph.D.
Committee Member

Shengfan Zhang, Ph.D.
Committee Member

ABSTRACT

Image classification is a sub-field of computer vision that focuses on identifying objects within digital images. In order to improve image classification we must address the following areas of improvement: 1) Single and Multi-View data quality using data pre-processing techniques. 2) Enhancing deep feature learning to extract alternative representation of the data. 3) Improving decision or prediction of labels. This dissertation presents a series of four published papers that explore different improvements of image classification. In our first paper, we explore the Siamese network architecture to create a Convolution Neural Network based similarity metric. We learn the priority features that differentiate two given input images. The metric proposed achieves state-of-the-art F_β measure. In our second paper, we explore multi-view data classification. We investigate the application of Generative Adversarial Networks GANs on Multi-view data image classification and few-shot learning. Experimental results show that our method outperforms state-of-the-art research. In our third paper, we take on the challenge of improving ResNet backbone model. For this task, we focus on improving channel attention mechanisms. We utilize Discrete Wavelet Transform compression to address the channel representation problem. Experimental results on ImageNet shows that our method outperforms baseline SENet-34 and SOTA FcaNet-34 at no extra computational cost. In our fourth paper, we investigate further the potential of orthogonalization of filters for extraction of diverse information for channel attention. We prove that using only random constant orthogonal filters is sufficient enough to achieve good channel attention. We test our proposed method using ImageNet, Places365, and Birds datasets for image classification, MS-COCO for object detection, and instance segmentation tasks. Our method outperforms FcaNet, and WaveNet and achieves the state-of-the-art results.

DEDICATION

This dissertation is dedicated to the memory of my beloved father, **Kanaan Hadi Salman**, who passed away Dec. 28^{th}, 2022. My father was not only my parent but also my mentor, my role model, and my inspiration. He instilled in me a love for learning and a strong work ethic from a young age. He always encouraged me to pursue my dreams and never gave up on me, even when I faced challenges and setbacks. Although he is no longer with us, his legacy lives on through me and my academic achievements. I know that he is proud of me for completing this dissertation and earning my PhD degree. I will always cherish the memories of the time we spent together and the valuable lessons he taught me. I hope that this dissertation serves as a fitting tribute to his memory and a testament to his influence on my life. Rest in peace, baba. You will forever be missed but never forgotten.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to very important people in my life: my mother and my sisters. Without their unwavering support, encouragement, and love, I would not have been able to complete this journey and achieve my PhD.

My mother, Maha Moghazy, has always been my rock and my source of inspiration. She has sacrificed so much for me and has always believed in me, even when I didn't believe in myself. Her guidance, wisdom, and unwavering support have been instrumental in helping me navigate through the challenges of pursuing a PhD. I am forever grateful for her love and sacrifice.

I would also like to acknowledge my sister, Laila Salman, for her endless support and encouragement. She has been my support, my sounding board, and my confidant throughout this entire process. Her unwavering faith in me and her ability to push me to be my best have been critical to my success. I am truly blessed to have her in my life.

Last, I would like to acknowledge my sister, Safa Salman, for her care and guidance through ups and downs.

In conclusion, I owe a debt of gratitude to my mother and sisters that can never be fully repaid. I hope to one day make them proud of me and to live up to the high expectations they have set for me. Thank you for everything.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PUBLICATIONS

Chapter 2 H. Salman and J. Zhan, ”Similarity Metric for Millions of Unlabeled Face Images,” 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 1033-1040.

Chapter 3 H. Salman and J. Zhan, ”Semi-Supervised Learning and Feature Fusion for Multi-view Data Clustering,” 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 645-650

Chapter 4 H. Salman, C. Parks, S. Y. Hong and J. Zhan, ”WaveNets: Wavelet Channel Attention Networks,” 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 1107-1113.

Chapter 5 H. Salman, C. Parks and M. Swan, ”OrthoNets: Orthogonal Channel Attention Networks,” in press.

1 Introduction

In recent years, there has been an exponential increase in the quantity of visual data created and shared online. Images and videos are now the predominant form of media on social media platforms, and the ability to automatically comprehend these visual data has become a crucial task for a variety of applications [1], such as image classification, object recognition, face recognition, autonomous vehicles, and medical imaging.

Image Classification is the process of assigning it to one or more predetermined classes or categories. The field of artificial intelligence that focuses on developing algorithms that can learn from data and make predictions or judgments based on this learning is known as machine learning. As showed in figures 1.1, 1.2 There are a variety of machine learning algorithms, such as supervised learning, unsupervised learning, and reinforcement learning. In the context of image classification, machine learning algorithms are taught using a tagged image dataset. During training, the algorithm learns to recognize the characteristics that describe each category and to make accurate predictions on unseen data. The most used technique for image classification is supervised learning, in which a training dataset of labeled photos is used to teach a machine learning algorithm to recognize the patterns and features associated with each class. After training, the algorithm can be applied to new photos to categorize them into one of the previously defined categories. Common image categorization machine learning algorithms include convolutional neural networks (CNNs), decision trees, and support vector machines (SVMs). CNNs are a specialized sort of neural network that has demonstrated exceptional effectiveness in image classification tasks due to their capacity to automatically learn features from raw picture data. These networks use convolutional layers to extract local image features, and fully connected layers to make the final classification determination. However, the majority of existing image classification methods assume that images are single-view, or viewed from a single angle or perspective. In numerous real-world situations, images may be captured from multiple perspectives or contain multiple objects, making them multi-view. Multi-view images present a unique classification challenge because

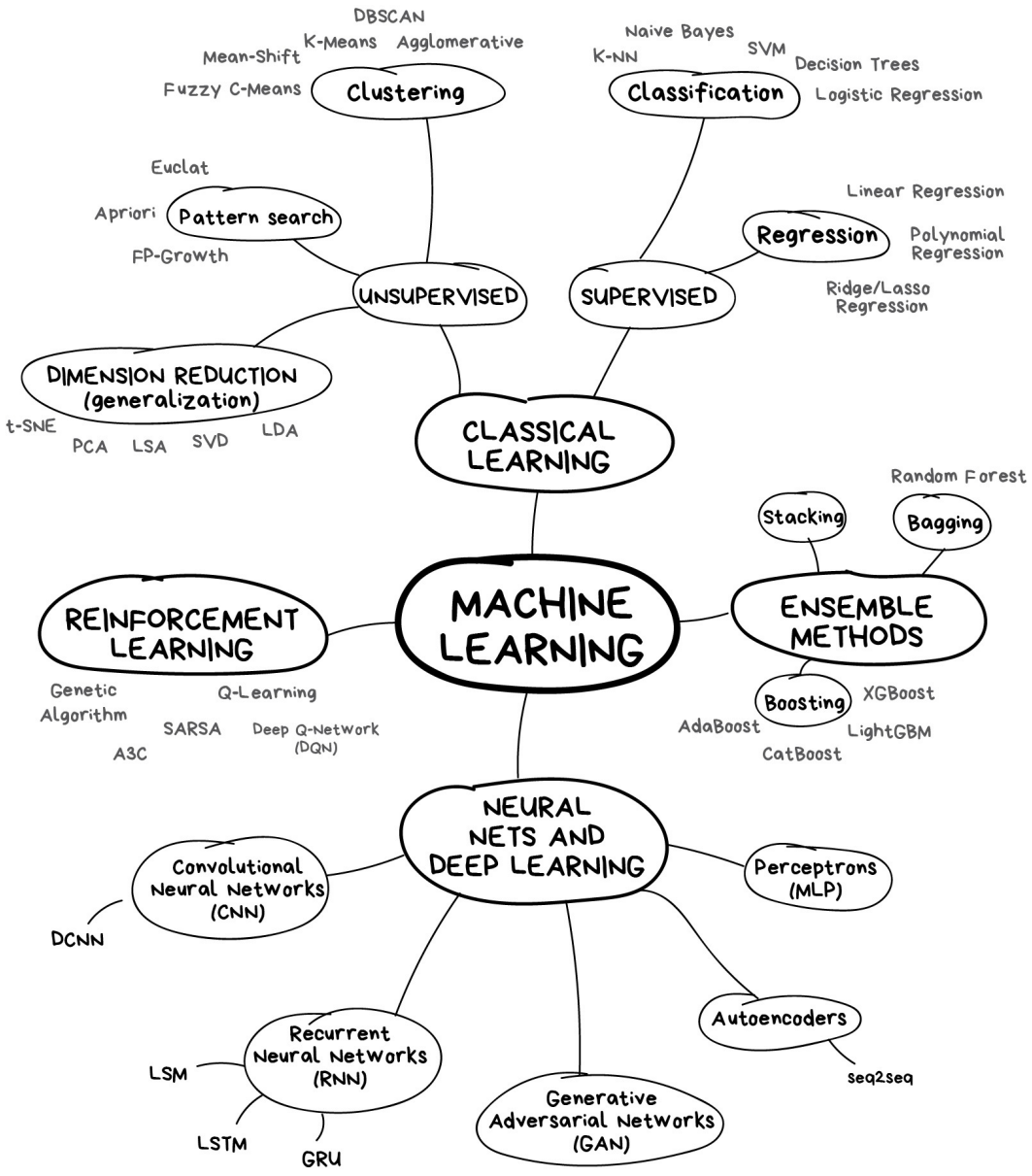

Figure 1.1: Machine Learning [2]

they require the network to recognize and integrate information from multiple perspectives or objects. In this thesis, we investigate the classification of single- and multi-view images using neural networks. We investigate various CNN architectures that can handle both single-view and multi-view images. Additionally, we investigate a wide range of information enhancement techniques that can improve the efficiency of these n etworks. We evaluate the performance of these methods on various single-view and Multi-View datasets. Overall, our research contributes to the field of

computer vision by developing new techniques for classifying single-view and multi-view images using neural networks. The findings of this study have significant implications for a wide range of applications, from self-driving cars to medical imaging, where single and multi-view images are gaining popularity.

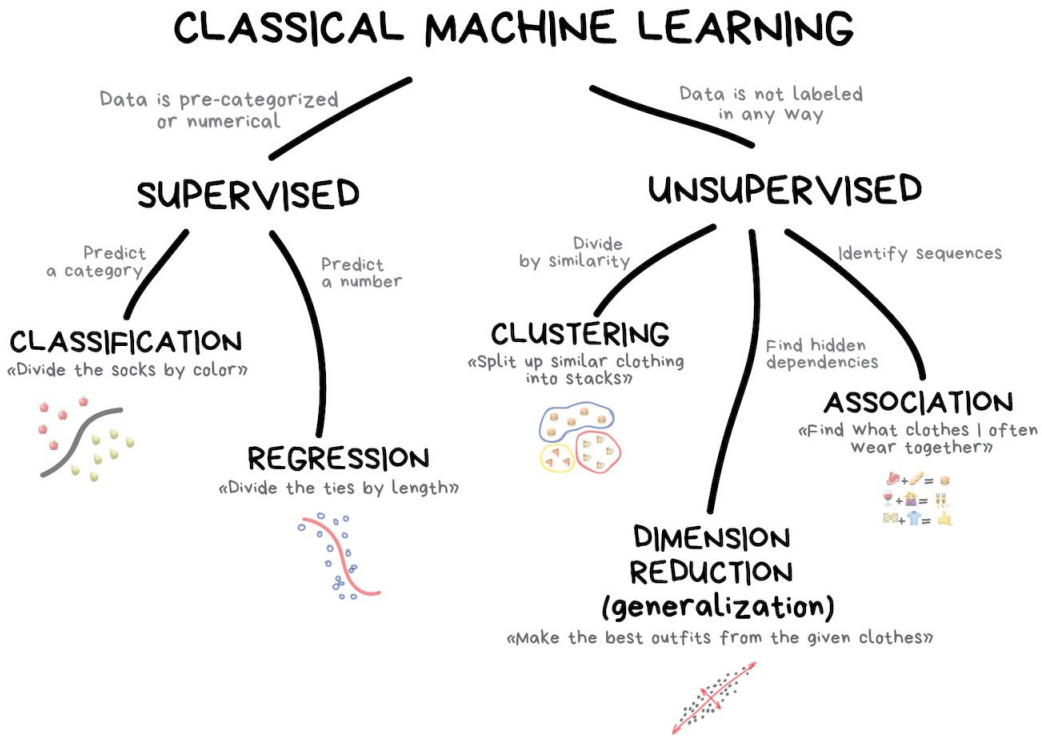

Figure 1.2: Types of classical machine learning [2]

1.1 Research Objectives

In this dissertation, we will be concerned with machine learning based logical units specially deep learning approaches.

We start by investigating the potential of using Siamese network for the purpose of learning a similarity metric to classify large dataset of faces. With the success of this work, we were motivated to explore multi-view data classification. Highly influenced by the technique of Generative Adversarial Networks (GANs), we explore the utilization of GANs for the classification of multi-view data, thereby creating multi-view GAN (MVSGAN). After completion of the previ-

ously mentioned work, we gained a deeper understanding of Neural Networks and were compelled to dig deeper at improving backbone networks like ResNet. We were motivated by the success of the backbone in multiple deep learning tasks like classification and segmentation. We look into add-on module improvement specifically channel attention add-on modules. We were able to improve on current methods and become state of the art as discussed in WaveNet and OrthoNet papers.

1.2 Dissertation Contributions

In this dissertation we present four research papers that address key problems in image classification, detection an segmentation using deep learning:

- In chapter two we discuss big data clustering and potential of Siamese Network as a similarity metric for millions of unlabeled face images. We first explore the Siamese network architecture to create a Convolution Neural Network (CNN) based similarity metric. We learn the priority features that differentiate two given input images. The metric is trained on MSCeleb (10M faces of 100k individuals) and VGGFace2 (3.3M faces of 9k individuals) then tested using LFW (13K faces of 5,749 individuals), MSCeleb, and VGGFace2 datasets. The network is evaluated through F_β measure and accuracy. The metric proposed in this paper achieves state of the art F_β measure of 0.94 for LFW benchmark. This work is described in "Similarity Metric for Millions of Unlabeled Face Images".

- In chapter three, we investigate the application of Generative Adversarial Networks on Multi-View data clustering and Few-Shot learning. We explore multi-view data classification. Nowadays, most of the data have multiple views and each view emphasizes a unique feature set of the data. We investigate the application of Generative Adversarial Networks GANs on Multi-view data for the task of clustering and few-shot learning. We propose mvSGAN, a deep learning approach to GAN multi-view clustering, where generator and classifier networks are in a competitive min-max game. A multi-view learning algorithm is implemented

with a mini-batch which can handle large data sets. We test the accuracy of our method in clustering real-world data sets. The experimental results show that our method outperforms state-of-the-art research. This work is described in "Semi-Supervised Learning and Feature Fusion for Multi-view Data Clustering".

- In chapter four, we theoretically and experimentally investigate the potential of using discrete wavelet transform as a lossy compression for channel attention mechanisms in deep convolutional neural networks. We further explore improving benchmark ResNet to achieve better image classification accuracy. Channel attention is used to learn to emphasize the importance of certain channels over other channels in deep neural networks. Designing effective channel attention mechanisms requires finding a solution to enhance feature preservation and diversification in modeling channel inter-dependencies. we utilize the Wavelet transform compression to address the channel representation problem. We first test Haar Wavelet transform as a replacement of GAP. We prove that global average pooling is equivalent to the recursive approximate Haar wavelet transform. With this proof, we generalize channel attention using Wavelet compression for feature preservation and name it WaveNet. We utilize different wavelets for extraction of channel attention. We test our proposed method using ImageNet benchmark on tasks of classification. Out method outperforms baseline SENet and state-of-the-art FcaNet on ResNet-34 at no extra computational cost. This work is described in "WaveNets: Wavelet Channel Attention Networks".

- In chapter five, we explore the orthogonal property for implementation of channel attention mechanism. We explore technique for reduction of training parameters in channel attention in deep convolutional neural networks. we investigate the potential of orthogonalization of filters for extraction of diverse information for channel attention. In this work, we demonstrate the effect of diversification of channel information on channel attention. We prove that using only random constant orthogonal filters is sufficient enough to achieve good channel attention. We prove that FCANet has the orthogonal property by definition. With this proof,

we generalize channel attention using standard 2D convolution with random orthogonal filter initialization and name it OrthoNet. Due to randomness we execute each experiment at least 5 times and record the average of results. Implementation of our method can be embedded within existing channel attention methods with a couple of lines of code. We test our proposed method using ImageNet, Places365, and Birds datasets for image classification, MS-COCO for object detection, and instance segmentation tasks. Our method outperforms the baseline SENet, and achieves the state-of-the-art results with acceptable margin of error. This work is described in "OrthoNets: Orthogonal Channel Attention Networks".

1.3 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 presents a similarity metric for face image labeling in an effort to explore Siamese neural networks. Next, Chapter 3 focuses on multi-view data clustering using semi-supervised learning and feature fusion. The following Chapter 4 and 5 explore improving channel attention mechanisms theory and implementation. Finally, Chapter 6 concludes the dissertation and offers some potential future works.

References

[1] *Image Analysis and Processing – ICIAP 2019: 20th International Conference, Trento, Italy, September 9–13, 2019, Proceedings, Part II*, Trento, Italy: Springer-Verlag, 2019, ISBN: 978-3-030-30644-1.

[2] *Machine learning for everyone*, `https://vas3k.com/blog/machine_learning/`, Accessed: 2010-09-30.

2 Chapter 2

Similarity Metric for Millions of Unlabeled Face Images

Hadi Salman, Justin Zhan

Abstract– Lately, image classification with unknown labels has been the focus of many research related to social media and cyber security. In order to cluster images into labels, a similarity metric is used to compare images and based on the output a decision is made. In this article, we propose a Convolution Neural Network (CNN) based metric that achieves state of the art results. The proposed approach learns the priority features that differentiate two given input faces. The metric is trained on MSCeleb (10M faces of 100k individuals) and VGGFace2 (3.3M faces of 9k individuals) then tested using LFW (13 K faces of 5,749 individuals), MSCeleb, and VGGFace2 datasets. The network is evaluated through F_β measure and accuracy. The metric proposed in this paper achieves state of the art F_β measure of 0.94 for LFW benchmark.

2.1 Introduction

Computer vision and machine intelligence has become of great importance in most of the state of the art technological achievements. One very popular technique used for Big Data vision-related problems is convolutional neural network (CNN) [1]–[4]. Mimicking brain behavior, a set of training data is supplied to the network to tweak the activation functions. After training, the network can be used to make decisions to match the query to the training sample closest to it. CNNs proved very effective for face recognition given training data with known relative identities [3]–[5]. Unrestricted Face recognition and identity clustering are often two sides of the same coin. In order to categorize a group of faces by identity, accurate recognition classifier has to be implemented that is trained on similar known identities. As an example, with the use of CNN, labeled faces in the wild (LFW) benchmark has been significantly improved from 97% [5] to 99% [4], [5]. The increase in accuracy can be further optimized by several approaches that include

increasing and diversifying the training data [6], enhancing noisy features [7] and post clustering identities based on featured sets [8]–[11].

Many studies have discussed clustering collection of face pictures using a Rank-Order approach [11], [12]. In this technique, images are passed through a trained feature extraction method. The, a nearest neighbor classifier is applied to those images with respect to each other. The neighbor list is then used to rank the similarity between two images. In practice, the number of identities k can be different for each identity cluster. Also, applying equal weights in the election process for the extracted features may result in lowering the cluster quality.

The purpose of this paper is to approach the task of face image clustering as a binary face difference metric. The overall approach is to read in two face images and decide if both faces belong to the same person. Such task comes naturally to humans which was enough motivation to apply deep learning and convolutional neural networks in an effort to achieve high accuracy.

The rest of this paper is structured as follows. Section 2.2 discusses previous work in feature extraction, clustering and classification techniques related to face identity clustering. Section 2.3 outlines the mathematical background, the architecture of the convolutional neural network used in this experiment. Section 2.4 presents the evaluation of the network performance using accuracy and F-measure applied on LFW [13] , MSCeleb [14], and VGGFace2 [15] benchmarks are also used. Section 2.5 concludes this work with recommendations for future work.

2.2 Related Works

In the field of image clustering, there isn't a unified image representation for all the implementations. Clustering depend on the features extracted just as much as the technique of clustering used. In this section, we briefly mention some of the approaches that have been used before to enhance face annotation/clustering.

2.2.1 Feature Extraction Methods

The abstract concept of clustering images is to compute the similarity between two images with respect to a precalculated threshold. By definition, the more information (features) acquired from the image the more accurate the similarity value which will yield quality clusters. SURF [16] Speed-up Robust Features has been proved very effective in face annotation as demonstrated by Edwin et al. [17]. It takes into consideration different image conditions like blur and illumination. Another technique is the Gabor feature described by Kamarainen et al. [18]. The main drawback in this technique is that the Garbor filter does not have a representation for face shapes which makes it a bad candidate for face annotation. Vrushali et al. [19] utilize Scale Invariant Feature Transform SIFT [20] to achieve face recognition. There are four main steps in SIFT: Scale-space detection, key point localization, orientation assignment, and key point descriptor. One drawback to this technique is that it uses contouring for feature extraction which can easily be incorrect in cases of illumination and shadows. Another technique demonstrated by Jamil et al. [21] uses dates and cloths colors as extra information to the image to be able to have more accurate clustering results. This concept relies heavily on the assumption that if two pictures of the same person was taken shortly after each other then the color of cloths and the timestamp should affect highly the decision of adding those two images to the same identity cluster. Lately, CNNs have been the number one choice for feature extraction due to their robustness and high accuracy peaking to humen levels. In DeepFace [4] CNNs were trained on 4.4 million face images and reached more than 97% accuracy on LFW [13]. The number of features extracted was 4096 per feature vector. Another CNN is FaceNet [5] which achieves 99.63% accuracy after training on approximately 200M face images with 8M face identities. Both approaches has been trained on categorical foreknown identities. the efficiency of both approaches drops significantly for cases with new identities being introduced in testing and low resolution input [22].

2.2.2 Image Clustering Techniques

There have been several implementations of spectral, hierarchical and rank-order clustering techniques. Ho et al. [23] utilized spectral clustering and assumed skin as lambertian surface to calculate similarity between two faces. Zhao et al. [24] utilized hierarchical clustering supported by a two-dimensional Hidden Markov Model for probability estimation of certain faces to appear together. Zhu et al. [25] described a multi-phase approach where the calculated dissimilarity is based on the ranking of the compared faces relative to the nearest neighbor list of each face. After ranking, hierarchical clustering is performed based on the new ranking order. Based on Zhu, Wang implemented an approximate k-NN graph construction method to build the nearest neighbor list. Vidal and Favaro [26] derived several subspaces from the data and used them to perform clustering for the assumption that selected subspaces represent the data efficiently. Wang et al. [27] proposed a face search system that utilized approximate k-NN supported by a commercial COTS matcher in a cascade framework. It achieved 98.2% accuracy on LFW [13]. Otto et al. [11] proposed an approach for clustering millions of faces by identity. It utilized the approximate rank-order clustering and k-mean classification which achieved F-measure of 0.87 on LFW [13] and 0.71 for the Youtube benchmark [28]. Yan et al. [29] proposed a fashion image deep clustering (FiDC) model which includes two parts, feature representation and clustering. The model focuses on integration of the learning process of the autoencoder and the clustering together. Mrabah et al. [30] proposed an unsupervised learning approach for untrained labels. The proposed approach solve a clustering-reconstruction trade-off by eliminating the reconstruction objective while preserving the space topology.

2.2.3 Classification Techniques

Clustering massive number of images is a complex task. For example, using DeepFace, the feature dimension vector is 4096. The complexity for such task requires a classification technique suited for high-dimensional data. To describe this further, we introduce the different kinds of classifiers that can be used for the task.

Linear Classifiers

A classifier is linear if classification areas are separated by a hyperplane. The feature space decision boundary is also a linear function. Examples like support vector machine SVM and logistic regression use linear decision boundaries. The hyperspace linear equation is used as a threshold for the mapped features space. While the use of linear classifiers is lower in calculation cost for high dimensional feature space, the clustering quality for big data requires more powerful techniques for the task of face image clustering.

Non-Linear Classifiers and Neural Networks

Images are supplied to a Convolutional Neural Network. The CNN produces a set of features known as image embedding. This features can be used to describe the image with less dimensions than the original image. When followed by a set of dense layers, the output can be reduced to lower dimentions based on the priority of the features.

In [31], Bukovcikova et al. proposed a Siamese NN approach for face recognition in uncontrolled conditions including poses, and illumination. Results indicate the efficiency of using this technique will increase by using large size of training set. Khalil-Hani et al. [32] proposed a face verification system that provided encouraging results for the task of face verification and encouraged the use of max pooling and fully connected layers for better classification accuracy. Another approach introduced by Huang et al. [33] target gender classification using Siamese network for feature extraction. In [34], Berlemont et al. proposed the use of Siamese Neural Network as a metric for initial gesture classification. The paper indicates the superiority of the nonlinear metric in cases of unknown gestures. Other works include

2.3 Methods

This section gives a brief overview of the approach starting with the image representation and training data generation. The aim of this research is to develop a metric that can be used to

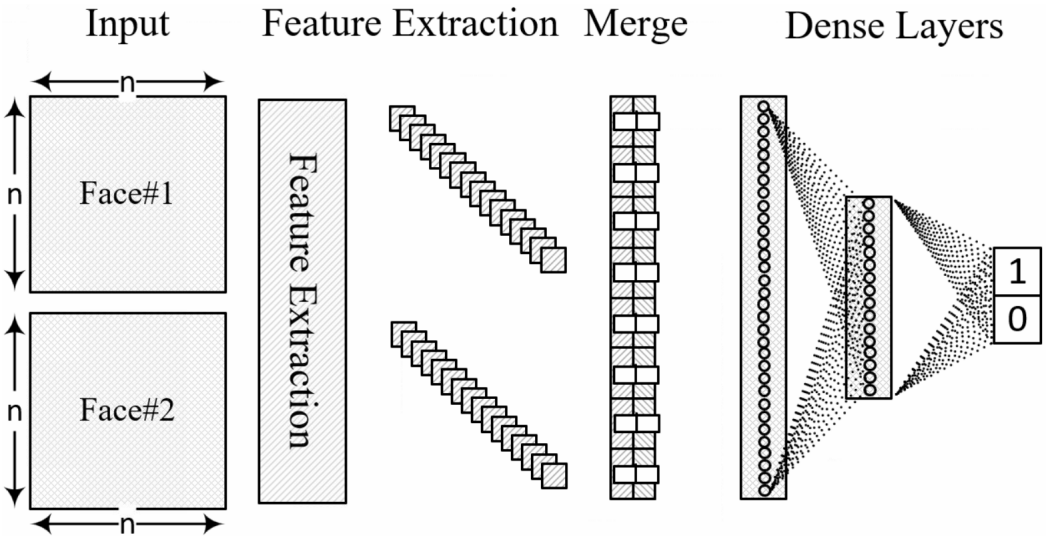

Figure 2.1: Siamese network architecture

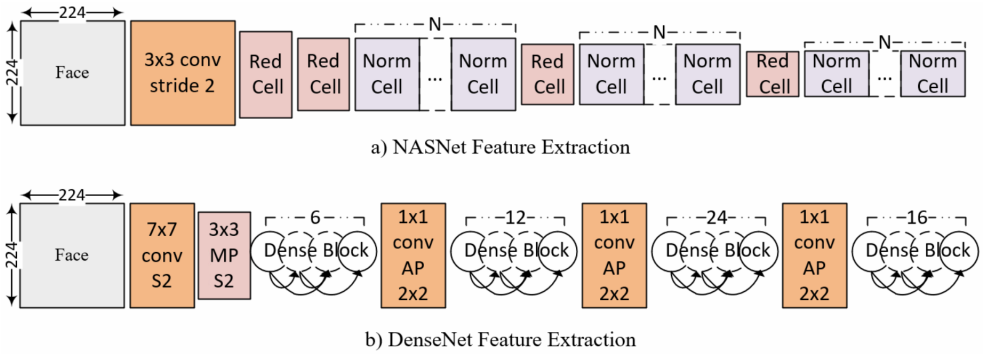

Figure 2.2: Deep Learning feature extraction

evaluate a probabilistic similarity between two face images based on feature extraction methods from NASNet [35] and DenseNet [36]. The metric takes an input of two face images and extracts the priority features to be used later for distance calculation. The metric is able to achieve state of the art accuracy for untrained labels. Further, when fine tuned on a relatively small set of testing data, the metric demonstrated high F_β compared to [11].

2.3.1 Data Preparation

The main goal of this research is to develop a Deep Convolutional Neural Network metric that takes in two face images with unknown identities and estimate the probability that they belong to the same person. Following research strategies mentioned in 2.2 we start with preparation of

data. For this task, the input images are passed through a landmark detection algorithm. In this work, we utilize the pre-aligned face images provided by datasets. Images are paired to either faces that share the same identity or other different identities. We maintain a portions of 48% positive pairs having face images that belong to the same person and 52% are negative pairs where the face images belong to different people. One hot encoding was used to concatenate both images and a label that is either 1 or 0. As demonstrated in Algorithm 1, the approach of picking training and validation data is completely random in nature. The reason for that is to have potentially a full representation of the dataset identities while maintaining a small number of pairs in training compared to the total number of pairs in the dataset.

Algorithm 1 Data generation

Require: *IDs, dataset identities*, *S, the number of similarity comparisons per identity*, *D, the number of different identity comparisons.*
Ensure: *SD, serialized one hot encoding data*

```
for id ∈ IDs do
    idF ← listFaces(id)
    randFa ← random.choices(idF,k = S)
    randFb ← random.choices(idF,k = S)
    for i ← 0 to S do
        SD.append(randFa[i],randFb[i],True)
    end for
end for
for id ∈ IDs do
    idFa ← listFaces(id)
    remIDs ← IDs − id
    randIDs ← random.choices(remIDs,k = D)
    for rid ∈ randIDs do
        idFb ← listFaces(rid)
        randFa ← random.choice(id)
        randFb ← random.choice(rid)
        for i ← 0 to D do
            SD.append(randFa[i],randFb[i],False)
        end for
    end for
end for
return SD
```

2.3.2 Feature Extraction Model

The recipe to high quality results is the accuracy of feature extraction model prior to classification. Neural network based feature extraction offers the capability to learn the best filters to

fit the problem and use that to produce high quality task relative feature extraction. NASNet [35] is a search algorithm that uses a set of predefined network blocks to find the best combination that yields As showed in the general approach in Figure 2.1, the Data generated from Algorithm 1 is fed to a two column network. Both columns are instances of the feature extraction layers adopted from NASNet and DenseNet showed in Figure 2.2. The input of the feature extraction is $224 * 224$ images with 3 color channels for NasNet and DenseNet.

2.3.3 Classification

For the sake of the problem, the output features extracted from the two column network are flattened and subtracted to create a feature tensor. There are two different methods for classification that are used for each network. The first method is a set of sequential dense layers used to map the non-linear function between the difference features tensor and the binary labels 1 and 0. The fist 2 dense layers uses relu function to drop the feature size to 1024 then to 512. The last layer uses sigmoid function for activation to represent the output as a number between the labels 0 and 1. The Second method is *k*-nn based classification. In this method we train the classifier on the same data after training the network feature extraction layers. The classifier is used to perform a nearest neighbor election based clustering where the number of neighbors is chosen by experiment.

Algorithm 2 Pairwise clustering

Require: *Images, face images in the wild*, TM, *the trained model for FaceDiff.*
Ensure: *CD, clustering dictionary of images*

```
Cid ← 1
for Fa ∈ Images do
    CD[Fa] ← Cid
    for Fb ∈ Images do
        L ← TM.predict(Fa,Fb)
        if L == true then
            CD[Fb] ← Cid
            Images.remove(Fb)
        end if
    end for
    Images.remove(Fa)
    Cid ← Cid + 1
end for
return CD
```

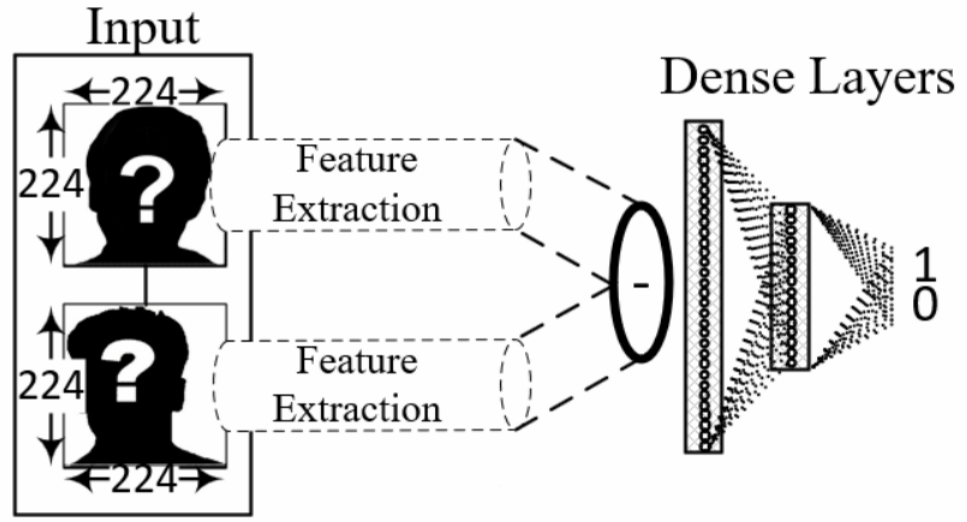

Figure 2.3: Dense layers classification

2.3.4 Clustering

In order to achieve better clustering accuracy, the trained network is used for pairwise evaluation. Given a set of images, the goal is to find the images that share the same identity. The initial basic approach is to compare each image to the remaining images which will be of complexity $O(n^2)$. The good news is that we can run this task in parallel.

2.4 Experiments and Results

The target of the experiment is to study the capability of using deep learned feature extraction models and classifiers for the task of clustering. This section first describes the experiments to learn the best feature extraction leading to state of the art clustering results for MS-Celeb-1M, VGGFace2, and LFW. Each network is trained on Faces from MS-Celeb-1M or VGGFace2. The Networks are then tested on all three datasets. Each network is then fine tuned using LFW and retested on LFW. Evaluation of the network is presented using the F_β scores, and Fraction Correct (FC). The evaluation criteria are discussed in section 2.4.2.

2.4.1 Setup and Parameters

This section describes each network feature extraction and classification parameters. There are mainly 4 networks based on 2 feature extraction methods. The networks are trained on MS-Celeb-1M with total of 1M pairs. Another instance of the networks is trained on VGGFace2 with total of 500k pairs. Training data are picked using algorithm 1 with a near equal number of pairs per label. Each instance of the networks is tested on LFW, VGG, and MS-Celeb-1M.

NasNet FaceDiff-N

Feature extraction using NASNet showed if Figure 2.2-a has two types of cells that are arranged sequentially. The main difference between normal and reduction cells is the output feature map size. Both cells are learned by the search controller Recurrent Neural Network RNN. The controller has the choice of applying an operation on a hidden layer and choosing the optimal merge operation between two hidden layers. There are B blocks each has 5 prediction steps made by 5 different softmax classifiers [35]. The steps produces next hidden state from 2 selected past hidden states. The choice of number of blocks B is manually chosen depending on the experiment. The number of initial convolutional filters and the reputations N for Normal cells are free parameters that are tailored to the scale of the problem [35].

DenseNet FaceDiff-D

The input image is padded with zersoe to be of size (230,230,3) then passed through a 2D Convolution layer yeilding a tensor of size (112,112,64).The tensor is passed through a Batch-Normalization layer then a relu activation layer where size of tensor does not change. A layer of 2D Max pooling is performed to reach size of (56,56,64) per image. A series of Dense Layers and 1x1 2D Convolution Layers are performed each with 6, 12, and 24 sub layers consisting of BatchNormalization, relu Activation, and 2D Convolution Layers as showed in Figure 2.2-b.

Table 2.1: Datasets statistics

Description	images	total identities	PW samples	testing	training
LFW	13,233	5749	87M+	5k	25k+
VGGFace2	3.3M+	9k+	5.4T+	50k	500k+
MS-Celeb-1M	10M	100k+	50T+	200k	1M+

2.4.2 Evaluation Metrics

In this section, we discuss the different metrics that give us a better understanding of the evaluation of the clusters.

F_β score

There are two main possibilities when clustering two images. Either the images belong together and were clustered together, also know as True Positive (TP). When assigning images that do not share the same identity to different clusters, it is know as True Negative (TN). There are two main errors that can happen in clustering. Either assign two images that do not share the same identity to the same cluster also know as False Positive (FP) or assign two images that share the same identity to different clusters also know as True Negative (TN). The harmonic mean of precision and recall is known as the (F_β) score which is calculated as showed in equation 2.1.

$$F_\beta = \frac{precision * recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{2.1}$$

Fraction Correct (FC)

Another measure for binary classification problems is the accuracy (Fraction Correct), which is the ratio between the number of correct classifications to the total number of classifications. FC measure can be calculated as showed in equation 2.2. The accuracy can be perceived as the ratio between True Positive and Negative to the total number of Pairs. If $FC = 1$, then all samples are classified correctly. If $FC = 0$, then all the samples are misclassified.

$$FC = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.2}$$

Table 2.2: FaceDiff-N, FaceDiff-D performance evaluation

Network	Training Set	Testing set	Accuracy	F-measure	TP	TN	FP	FN
FaceDiff-N (57,781,909 parameters)	MS-Celeb-1M (1M pairs) 5 epochs	MS-Celeb-1M (200k)	0.94	0.94	0.467	0.473	0.033	0.033
		VGGFace2 (50k)*	0.633	0.638	0.33	0.304	0.17	0.196
		LFW (5k)*	0.8	0.813	0.437	0.364	0.062	0.138
FaceDiff-D (58,944,065 parameters)	MS-Celeb-1M (1M pairs) 2 epochs	MS-Celeb-1M (200k)	0.922	0.923	0.479	0.443	0.021	0.057
		VGGFace2 (50k)*	0.661	0.67	0.4	0.212	0.1	0.288
		LFW (5k)*	0.788	0.8	0.448	0.34	0.051	0.161
	VGGFace2 (500,000) 6 epochs	MS-Celeb-1M (200k)*	0.772	0.775	0.486	0.235	0.014	0.265
		VGGFace2 (50k)	0.79	0.8	0.457	0.33	0.043	0.17
		LFW (5k)*	0.866	0.87	0.459	0.406	0.04	0.095

Binary Cross-Entropy Loss is a measure of performance of a models based on the probability of correct prediction. The loss function has low values when the label prediction accuracy is high and vice versa. The cross-entropy loss for label y, and probability of true prediction p is calculated as showed in Equation 2.3. The binary cross-entropy loss is equivalent to maximizing the log-likelihood in a two class model.

$$H(y,p) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{if } otherwise \end{cases} \tag{2.3}$$

2.4.3 Datasets

LFW

Labeled Faces in the Wild [13] is a database of face images designed to study the problem of unconstrained face recognition. There are 13233 image samples for a total of 5749 distinct identities. There are 1680 identities with more than one image sample. The total possible number of unique generated pairwise image comparisons for LFW is $(13233 * 13233)/2 = 87556144$ input instances that can be used for training or testing the models. The number of generated true positive pairs to the generated false positive pairs is highly bias thus, a random subset of sample is selected to achieve a ratio of $1:1$ for true positive and false positive pairs. LFW does not provide enough diverse samples per identity rendering it unfit for training. This dataset will be used for fine tuning

Table 2.3: FaceDiff-N, FaceDiff-D cross dataset training evaluation

Network	Training Set	FT training set*	Testing set	Accuracy	F-measure	TP	TN	FP	FP
FaceDiff-N (57,781,909 parameters)	MS-Celeb-1M (1M pairs) 5 epochs	VGGFace2 (500k)	VGGFace2 (50k)	0.842	0.85	0.454	0.388	0.046	0.112
		LFW (36k)	LFW (5k)	0.884	0.883	0.45	0.433	0.048	0.068
FaceDiff-D (57,781,909 parameters)	MS-Celeb-1M (1M pairs) 5 epochs	VGGFace2 (500k)	VGGFace2 (50k)	0.851	0.858	0.458	0.394	0.042	0.106
		LFW (36k)	LFW (5k)	0.868	0.864	0.421	0.447	0.078	0.053
	VGGFace2 (500k) 6 epochs	LFW (36k)	LFW (5k)	0.8	0.813	0.437	0.364	0.062	0.138

Table 2.4: FaceDiff-N, FaceDiff-D LFW testing evaluation

Network	Training Set	Testing set	Accuracy	F-measure	TP	TN	FP	FN
FaceDiff-N	MS-Celeb-1M	LFW	0.937	0.939	0.49	0.447	0.01	0.053
FaceDiff-D	VGGFace2	LFW	0.936	0.943	0.47	0.474	0.03	0.026

networks trained on VGGFace2 and MS-Celeb-1M.

VGGFace2

This dataset is a large-scale face recognition library. The images are downloaded from google image search engine and is described to have large variations in pose, age, illumination, ethnicity and profession [15]. The dataset has more than 9000 identities each having between 87 to 843 samples. The number of samples is more than 3.3 million images. the number of possible unique comparisons are $(3.3M * 3.3M)/2 \approxeq 5.4M$ input instances. Testing and training sets were chosen randomly guaranteeing an un-bias true to false pairs ratio.

MS-Celeb-1M

A challenge dataset created by Microsoft [14] with one million celebrities in the real world. The dataset has 10M face images. There are nearly 100k distinct identities. The identites are from different countries and ethnicity. The number of samples is more than 50 trillion pairs. A summery of the datasets statistics is showed in Table 2.1.

2.4.4 FaceDiff-N Parameters and Evaluation

The network has 2 inputs images of size (224,224,3). The images are passed through NASNet feature extraction model yielding a pair of (7,7,1056) tensors and a total of 42,69,716 trainable parameters. The output tensor are then flattened and subtracted ending up with a feature tensor of size 51,744. The extracted feature list is then passed to a non-linear dense Layer to reduce the number of features to 1024. FaceDiff-N is trained with two more Dense Layers to reach an output size of size 1 representing a number between 0 and 1 with a default threshold of 0.5. After training, The FaceDiff-N network weights are used with a k-nn classifier replacing the Dense layers. The classifier is also trained with the same training data used in training the feature extraction. The number of neighbors is decided relative to the dataset training size. The network was trained using MS-Celeb-1M dataset for 5 epochs. FaceDiff-N achieves accuracy of 0.94 and f-measure of 0.94. The network successfully classified 93.4% of true positive pairs and 94.6% of true negative pairs. The trained network (FaceDiff-N-1M) is tested on random sample pairs extracted from LFW without fine tuning on LFW. The network achieves accuracy of 0.8, F_β of 0.82. The network was able to classify 87.4% of true positive pairs and 72.8% of true negative pairs. When tested on VGGFace2 without any fine tuning the network was able to classify 66% of True Positive samples and 60% of True Negative pairs.

2.4.5 FaceDiff-D Parameters and Evaluation

The network input pairs of images are of size (224,224,3). The pairs features are extracted using DenseNet feature extraction model. The ouput of the model is a pair of (7,7,1024) with a total of ..... trainable parameters. Following the same post feature extraction method, the tensors are flattened and subtracted to generate the difference feature tensor. A series of dense layers are trained to map the feature tensor to the 1/0 class output. When trained on MS-Celeb-1M dataset for 2 epochs, FaceDiff-D achieved accuracy of 0.92 and F_β of 0.92. FaceDiff-D was able to classify approximately 98% of true positive pairs and 89% of true negative pairs. When tested on VGGFace2 without fine tuning, the network was able to classify 80% of the true positive samples

Clustering Algorithm	F-measure
k-Means	0.36
Spectral (Euclidean)	0.2
Spectral (Approx. ROD)	0.43
Hierarchical	0.005
Rank-Order	0.65
Approx. Neighborhood Rank-order	0.83
Approx. Rank-Order	0.87
FaceDiff-N (Proposed Approach I)	0.939
FaceDiff-D (Proposed Approach II)	0.943

Table 2.5: LFW benchmark F_β comparison

while the network ability to classify true negative pairs was poor compared to other datasets tests. FaceDiff-D was also tested on LFW without fine tuning and achieved accuracy of 0.788 and F_β of 0.80. The network was able to classify 89.6% of true positive pairs and 68% of the true negative pairs. FaceDiff-D is also trained on VGGFace2 dataset for 6 epochs and then tested on MS-Celev-1M without fine tuning reaching accuracy of 0.772 and F_β of 0.775. The network was able to detect 96.4% of True Positive pairs. On the other hand the network was not able to perform well for true negative pairs. When tested on VGGFace2 test set, FaceDiff-D achieved accuracy of 0.79 and F_β of 0.80. The network was successfully able to classify 91.4% of true pairs and 66% of the true negative pairs. FaceDiff-D was tested on LFW without fine tuning and achieved accuracy of 0.866 and F_β of 0.87. The network was able to classify almost 92% of the true positive pairs and 81.2% of the true negative pairs.

2.4.6 Cross Dataset training FaceDiff-N and FaceDiff-D

FaceDiff-N and FaceDiff-D achieve high accuracy and F_β when trained on MS-Celeb-1M. When tested directly on LFW and VGGFace2, accuracy went down due to the differences between training dataset and testing dataset. In order to achieve better results, both networks were fine

tuned using LFW and VGGFace2 training sets described in Table 2.1. For FaceDiff-N, The trained network is loaded and further trained on LFW 25k pairs. FaceDiff-N achieves accuracy of 0.88 and F_β of 0.879. The network was able to correctly classify 89% of True positive pairs and 87% of the true negative pairs as showed in Table 2.3. The network was also fine tuned using VGGFace2 training set. Testing results show a significant increase in accuracy and F-measure. The network achieves accuracy of 0.842 and F_β of 0.85. The network was able to correctly classify 90.8% of true positive pairs and 77.6% of true negative pairs. FaceDiff-D trained originally on MS-Celeb-1M was fine tuned using VGGFace2 and LFW training sets. As showed in Table 2.3, the accuracy has increased to 0.868 for LFW and 0.85 for VGGFace2. The F_β score has also increased significantly reaching 0.864 for LFW and 0.858 for VGGFace2. The true positive pairs correctly classified for LFW and VGGFace2 are approximately 84% and 92% respectively.

Last, FaceDiff-D trained on VGGFace2 was fine tuned with LFW trainig set. The network achieved accuracy of 0.8 and F_β of 0.813. The true positive pairs correctly classified for LFW are approximately 87.4%.

2.4.7 LFW Benchmark Test

LFW dataset consists of $234,774$ true pairs. For the purpose of calculating the F-measure F_β, an equivalent number of negative pairs were randomly sampled from the dataset. The total number of positive and negative pairs is $469,548$ pairs. When tested using FaceDiff-N, accuracy of 0.937 and F_β of 0.939 is achieved as showed in Table 2.4. The network was able to correctly classify 98% of true positive pairs and 89.4% of true negative pairs showed in Table 2.3. FaceDiff-D was also tested on LFW benchmark and achieved state of the art accuracy of 0.936 and F_β of 0.943 compaired to its predecessors showed in Table 2.4. The network was able to correctly predict 94% of true positive pairs and 94.8% of true negative pairs showed in Table 2.3. Both networks achieved better results than conventional methods. FaceDiff-N performs better classifying true positive pairs and FaceDiff-D performs better for true negative paris as showed in Table 2.3.

2.5 Conclusion and Future Work

We have presented a novel deep learning approach for unknown identity face clustering. We show that the multi-class identity clustering problem can be transformed into a 1 class unknown identity classification problem where the main target of the network is to estimate the probability of both input images belong to the same identity. We demonstrate the superior performance of RNN based feature extraction architecture for face clustering. We demonstrate that better results can be achieved with more training data and the capability of generalization of architecture using cross dataset fine-tuning and testing. The proposed structure could be applied for other problems where linear metrics are inefficient such as image stitching and super resolution image similarity. We highly encourage further investigation of using FaceDiff-N and FaceDiff-D with probabilistic clustering algorithms.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[3] S. Liao, Zhen Lei, Dong Yi, and S. Z. Li, "A benchmark study of large-scale unconstrained face recognition," in *IEEE International Joint Conference on Biometrics*, Sep. 2014, pp. 1–8. DOI: 10.1109/BTAS.2014.6996301.

[4] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018, ISSN: 1556-6013. DOI: 10.1109/TIFS.2018.2833032.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[6] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014. arXiv: 1411.7923. [Online]. Available: http://arxiv.org/abs/1411.7923.

[7] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, May 2014, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2013.2292894.

[8] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient knn classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, May 2018, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2017.2673241.

[9] J. Zhao, J. Han, and L. Shao, "Unconstrained face recognition using a set-to-set distance measure on deep learned features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2679–2689, Oct. 2018, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2017.2710120.

[10] H. Ma, J. Gou, X. Wang, J. Ke, and S. Zeng, "Sparse coefficient-based *k* -nearest neighbor classification," *IEEE Access*, vol. 5, pp. 16 618–16 634, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2739807.

[11] C. Otto, D. Wang, and A. K. Jain, "Clustering millions of faces by identity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 289–303, Feb. 2018, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2679100.

[12] C. Zhu, F. Wen, and J. Sun, "A rank-order distance based clustering algorithm for face tagging," in *CVPR 2011*, Jun. 2011, pp. 481–488. DOI: 10.1109/CVPR.2011.5995680.

[13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, Oct. 2007.

[14] J. Harvey Adam. LaPlace. "Megapixels: Origins, ethics, and privacy implications of publicly available face recognition image datasets." (2019), [Online]. Available: https://megapixels.cc/ (visited on 04/18/2019).

[15] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *International Conference on Automatic Face and Gesture Recognition*, 2018.

[16] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[17] E. Paul and Ajeena Beegom A S, "Mining images for image annotation using surf detection technique," in *2015 International Conference on Control Communication Computing India (ICCC)*, Nov. 2015, pp. 724–728. DOI: 10.1109/ICCC.2015.7432989.

[18] J. Kamarainen, "Gabor features in image analysis," in *2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Oct. 2012, pp. 13–14. DOI: 10.1109/IPTA.2012.6469502.

[19] V. Purandare and K. T. Talele, "Efficient heterogeneous face recognition using scale invariant feature transform," in *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, Apr. 2014, pp. 305–310. DOI: 10.1109/CSCITA.2014.6839277.

[20] Zhu Daixian, "Sift algorithm analysis and optimization," in *2010 International Conference on Image Analysis and Signal Processing*, Apr. 2010, pp. 415–419. DOI: 10.1109/IASP.2010.5476084.

[21] N. Jamil and S. A. Sa'dan, "Automated face annotation for personal photo management," in *2014 International Conference on Computational Science and Technology (ICCST)*, Aug. 2014, pp. 1–5. DOI: 10.1109/ICCST.2014.7045176.

[22] M. Rani Golla and P. Sharma, "Performance evaluation of facenet on low resolution face images: First international conference, cnc 2018, gwalior, india, march 22-24, 2018, revised selected papers," in Jan. 2019, pp. 317–325, ISBN: 978-981-13-2371-3. DOI: 10.1007/978-981-13-2372-0_28.

[23] J. Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, Jun. 2003, pp. I–I. DOI: 10.1109/CVPR.2003.1211332.

[24] M. Zhao, Y. W. Teo, S. Liu, T.-S. Chua, and R. Jain, "Automatic person annotation of family photo album," in *Image and Video Retrieval*, H. Sundaram, M. Naphade, J. R. Smith, and Y. Rui, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 163–172, ISBN: 978-3-540-36019-3.

[25] C. Zhu, F. Wen, and J. Sun, "A rank-order distance based clustering algorithm for face tagging," in *CVPR 2011*, Jun. 2011, pp. 481–488. DOI: 10.1109/CVPR.2011.5995680.

[26] R. Vidal and P. Favaro, "Low rank subspace clustering (lrsc)," *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014, ICPR2012 Awarded Papers, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2013.08.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865513003012.

[27] D. Wang, C. Otto, and A. K. Jain, "Face search at scale," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1122–1136, Jun. 2017, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2582166.

[28] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR 2011*, Jun. 2011, pp. 529–534. DOI: 10.1109/CVPR.2011.5995566.

[29] C. Yan, U. S. Malhi, Y. Huang, and R. Tao, "Unsupervised deep clustering for fashion images," in *Knowledge Management in Organizations*, L. Uden, I.-H. Ting, and J. M. Corchado, Eds., Cham: Springer International Publishing, 2019, pp. 85–96.

[30] N. Mrabah, N. M. Khan, and R. Ksantini, "Deep clustering with a dynamic autoencoder," *CoRR*, vol. abs/1901.07752, 2019. arXiv: 1901.07752. [Online]. Available: http://arxiv.org/abs/1901.07752.

[31] Z. Bukovčiková, D. Sopiak, M. Oravec, and J. Pavlovičová, "Face verification using convolutional neural networks with siamese architecture," in *2017 International Symposium ELMAR*, Sep. 2017, pp. 205–208. DOI: 10.23919/ELMAR.2017.8124469.

[32] M. Khalil-Hani and L. S. Sung, "A convolutional neural network approach for face verification," in *2014 International Conference on High Performance Computing Simulation (HPCS)*, Jul. 2014, pp. 707–714. DOI: 10.1109/HPCSim.2014.6903759.

[33] Y. Huang, S. Liu, J. Hu, and W. Deng, "Metric-promoted siamese network for gender classification," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, May 2017, pp. 961–966. DOI: 10.1109/FG.2017.119.

[34] S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia, "Siamese neural network based similarity metric for inertial gesture classification and rejection," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 1, May 2015, pp. 1–6. DOI: 10.1109/FG.2015.7163112.

[35] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8697–8710. DOI: 10.1109/CVPR.2018.00907.

[36] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

3 Chapter 3

Semi-Supervised Learning and Feature Fusion for Multi-view Data Clustering

Hadi Salman, Justin Zhan

Abstract– Generative Adversarial Networks GANs have become widely used in Single-view classification tasks. Nowadays, most of the data have multiple views and each view emphasizes a unique feature set of the data. In this paper, we investigate the application of GANs on Multi-view data for the task of clustering and few-shot learning. We propose mvSGAN, a deep learning approach to GAN multi-view clustering, where generator and classifier networks are in a competitive min-max game. A multi-view learning algorithm is implemented with a mini-batch which can handle large data sets. We test the accuracy of our method in clustering real-world data sets. The experimental results show that our method outperforms state-of-the-art research.

3.1 Introduction

Different sensors and feature extraction methods,for the same object, have been widely used in areas like: user authentication, object classification, and visual speech recognition. Data gathered from multiple sensors and feature extraction methods are known as Multi-view (MV) data. Depending on the object and the method used, MV data vary in feature size and the complexity of the information extracted. For example, a patient can have multiple image scans which are considered views; a text description of the symptoms can also be a view of the data. Merging those different views together may yield more accurate, reliable and safer diagnosis than a single view. Since the data views complement each other, MV data fusion is an essential tool that is worthy of analyzing to achieve better object classification. There are many tasks that require labeling unlabeled data. The most common used task in computer vision is image clustering. In this task, the target uses a set of labeled data to learn an intermediate embedding that can be used to find the labels of unlabeled data. Clustering requires comparing object embedding together to find

the similarity between objects. Those methods have been widely used in social media analysis, computer vision and other fields [1]–[3]. There have been many studies on the extension from single view to multi view clustering. One very common approach for multi-view clustering learning methods is to build upon the traditional single view clustering method and adapt it to multi-view datasets. The most trivial method used is k-means based clustering. Other methods include, but are not limited to, hierarchical clustering, and spatial clustering, etc. Spectral clustering (SC) [4] is another representative clustering algorithm, that treats each data point as a node in a graph and thus transforming the problem into a graph-partitioning problem. Recently, there have been a number of proposed studies in the literature describing Multi-view Spatial Clustering (MVSC). This wave of literature reached its peak in 2019 describing in detail the relaxation solutions of graph balanced-cut problems.

In this paper, we investigate the extension of generative adversarial networks (GANs) [5] for multi-view semi-supervised learning over graphs. We present an enhanced extension of GraphSGAN that can achieve high accuracy results for multi-view data. mvSGAN maps each view's graph into an independent feature space then uses an election method to fuse the corresponding label views into one signal multi-class predictions. The proposed mvSGAN is tested on five different multi-view real world datasets. Experimental results show that mvSGAN highly outperforms several state-of-the-art methods including the latest Multi-view SpectralNet (MvSN) [6] method. mvSGAN is scalable and can be trained using mini-batch approach. Our contributions are as follows:

1. We introduce multi-view GANs as a optimal method to solve multi-view classification tasks on graphs using a semi-supervised approach.

2. We formulate a competitive game between view based generators and discriminators for mvSGAN. We analyze the training hyper-parameters and the theoretical working principles of the approach.

3. We evaluate our model on five benchmark real-world multi-view datasets. mvSGAN signif-

icantly outperforms previous works while maintaining scalability.

The rest of the paper is arranged as follows. In Section 3.2, we introduce the necessary definitions and related work. In Section 3.3.2, we present mvSGAN and discuss in details the working principals and execution methods of the design. In Section 3.4, we discuss in details the experiments conducted and show the superiority of our model in comparison with stat-of-the-art method. We close with a set of conclusions in Section 3.5.

3.2 Related Works

3.2.1 Basics of Graph-based Semi-Supervised Learning

Suppose that a set of n images are used for training. A small subset of size l are labeled. The remaining $u = n - l$ are unlabeled thus, cant be used for training. The goal of Semi-Supervised learning is to make use of the labeled images to estimate the labels for the unlabeled images which improves the quality of the network output.

In Multi-View, A set of different feature extraction techniques are used to compute a set of v views of the image represented by $({x_1}^{d_i}, \ldots, {x_n}^{d_i})$ and $i = 1, \ldots, v$. Each view has different dimensions d_i. In most of Big Data applications, the ratio of labeled images l to the unlabeled images u is very small. For a set of k classes of labels, one hot encoding is used to form $Y = [y1, \ldots, y_n]^T$ where the labeled images are introduced first as showed in equation 3.1.

$$Y = \begin{bmatrix} Y_l \\ Y_u \end{bmatrix} \tag{3.1}$$

In multi-view graph based model, a graph per view of the data is constructed. The graph is constructed based on the similarity between vertices. The similarity is usually calculated using the

exponential of the L2 norm showed in equation 3.2 or similar metrics.

$$A_{ij}{}^{v} = \frac{exp(-\|x_i{}^{v} - x_j{}^{v}\|_2{}^2)}{2\sigma^2} \tag{3.2}$$

Based on the similarity graph, an important concept is the Laplacian of the graph used to describe the connection status of the graph. The Laplacian is defined as $L = D - A$ where $A \in \{\mathbb{R}^{n \times n}\}$ is the adjacency matrix and D is the diagonal matrix where $D_i i = \sum_{j=1}^{n} A_{ij}$.

3.2.2 GFHF

The Gaussian Field and Harmonic Functions (GFHF) [7] is a single view method based on the assumption that provided set of labeled data are accurate and valid, the unlabeled ones can be propagated from the labeled one. The main goal of this cost function is to find the closest set of images and assigning the labels of the labeled ones to the unlabeled ones. The optimization cost function is as follows :

$$\min_{W} \mu \sum_{i=1}^{l} \|w_i - y_i\|_2^2 + \sum_{i,j=1}^{n} A_{ij} \left\|w_i - w_j\right\|_2^2 \tag{3.3}$$

3.2.3 MvSN

Multiview spectral clustering [6] using deep neural network for was proposed by Huang et. al. The overall approach of the research is to use SpectralNet [4] to learn the mapping of the multiview data to their fusion eigenvictors. The approach replaces the L2 affinity matrix with a Siamese Network that is learned for each view independently. The Siamese network provides a mapping function $\mathbf{y}_i = F_{\theta_{\text{Siamese}}}(\mathbf{x}_i)$ to a target space for each of he two inputs x_i the outputs are then subtracted to find the difference vector used to decide a probabilistic value of similarity. The

network weights are learned with a contrasting loss function showed in equation 3.4.

$$L_{\text{siamese}}(\theta) = \begin{cases} \|\mathbf{y}_i - \mathbf{y}_j\|^2 & \text{if}(\mathbf{x}_i, \mathbf{x}_j) \in D_{\text{pos}} \\ \max(0, l - \|\mathbf{y}_i - \mathbf{y}_j\|)^2 & \text{if}(\mathbf{x}_i, \mathbf{x}_j) \in D_{\text{neg}} \end{cases} \tag{3.4}$$

The affinity matrix is used to calculate the Laplacian which is used to learn embedding Z using equation 3.5.

$$\min_{\mathbf{Z}^T\mathbf{D}\mathbf{Z}=\mathbf{I}_{c\times c}} \text{Tr}(\mathbf{Z}^T(\mathbf{L})\mathbf{Z}) \tag{3.5}$$

The loss function is based on the assumption that points with large similarity should be clustered together. The fusion loss function used is showed in equation 3.6.

$$\mathscr{L}_{\text{fusion}}(\theta) = \sum_{k=1}^{v} \left(\alpha \|\mathbf{A}^k\|^2 + \frac{w_k}{(m^k)^2} \sum_{a,b=1}^{m^k} \mathbf{A}_{ab}^k \left\| \tilde{\mathbf{z}}_a^k - \tilde{\mathbf{z}}_b^k \right\|^2 \right) \tag{3.6}$$

There are two main drawbacks to this approach. The first concern is the computational cost training v independent Siamese networks and finding the orthogonal embedding for each network. The second is the fusion loss function uses the concept of linear combination whcih can be replaced with a neural network to find the nonlinear embedding thus improving the results.

3.2.4 GraphSGAN

In semisupervised learning we usually have the portion of labeled data V^L very small compared to unlabeled data V^U and the corresponding graph G as partially labeled graph [8].

GraphSGAN approach utilizes Generative Adversarial Networks (GANs) in which two models are trained such that one model (G) generates samples that best fit the training data and another model (D) is trained to distinguish real samples from fake ones generated by the other model. This process is mathematically described by a min-max game between the models. The loss function of such networks follow the following value function:

$$\min_G \max_D V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} \log[1 - D(G(\mathbf{z}))] \tag{3.7}$$

where p_d is the data distribution from the training data, $p_z(\mathbf{z})$ is a prior on input noise variables.

3.3 Model Framework and Proposed Approach

3.3.1 Architecture

For each input view v in the k views, we use an embedding network such as Line [9] and DeepWalk [10] to learn a latent embedding q_{vi} of fixed size. Each latent embedding q_{vi} is concatenated with the original feature w_{vi} as showed in equation 3.8

$$x_{vi} = q_{vi} + w_{vi} \tag{3.8}$$

The architecture of our model (mvSGAN) is showed in Figure 3.1. For each view of the data, an optimal generator G and a discriminator D is trained to fit a local solution. A discriminator is a classifier that has a stochastic probabilistic multi-class output. Each fully-connected layer in the classifier is followed by weight normalization to force parameter regularization. To achieve probabilistic outputs, the fully connected layers are followed by a softmax activation layer. The number of output classes is the number of original training classes and an extra class for the density gap. Once the Multi-view models are run we end up with probabilistic outputs for each view. The probabilistic outputs are non-linearly transformed using the Fusion layer to reach the final predication. The fusion layer is multi-layer perceptron network with a multiclass cross entropy loss function. The loss increases as the predicted probability diverges from the actual labels.

3.3.2 Multiview Model Learning Algorithm

The generator and discriminator are both represented by two loss functions which are mathematically differentiable with respect to input parameters. This allows the notwork to back-propagate for optimal weight estimation. For each view of data we have both localized Generative losses $\mathscr{L}_G(G,D)$ and Discriminative losses $\mathscr{L}_D(G,D)$. We define the losses to guarantee expected equilibrium in final clustering as follows:

$$\begin{aligned}\mathscr{L}_G &= \sum_{v=1}^{k} loss_{fm_v} + \sum_{v=1}^{k} \lambda_v loss_{pt_v} \\ \mathscr{L}_D &= \sum_{v=1}^{k} loss_{xl_v} + \sum_{v=1}^{k} \beta_v loss_{gan_v} + \\ &\quad \sum_{v=1}^{k} \gamma_v loss_{xu_v} + \sum_{v=1}^{k} loss_{pt_v}\end{aligned} \tag{3.9}$$

Generative Losses

In order to force samples to be generated in density gap, we have to enforce two main requirements:

1. For each view v, the generator G_v generates samples which are mapped into the central area of the current graph view. We introduce view based feature matching loss [11] as a solution to this requirement. $loss_{fm}$ aims to minimize the distances between generated samples and the average point of real samples. For simplicity we only care about the L_1 loss and the last layer of the neural network E therefore, a simplified loss function is showed in equation 3.10 and is used for each view independently.

$$\begin{aligned}loss_{fm} = &|\mathbb{E}_{\mathbf{x}_i \in X_{batch}} h^{(n)}(\mathbf{x}_i) - \\ &\mathbb{E}_{\mathbf{x}_j \sim G(\mathbf{z})} h^{(n)}(\mathbf{x}_j)|\end{aligned} \tag{3.10}$$

where $h^{(n)}$ is the last layer of the neural network E for feature matching.

2. Generated samples for each view v should **not** be over generated exactly at the center point to allow for spread of density gap samples. A simple solution to this requirement is to implement a repelling regularizer [12] which maintains production of samples that are not clustered in one or only few models of the batch data distribution p_{data}. To achieve required regularization, a pull-away term that runs at the representation level is implemented as showed in equation 3.11.

$$loss_{pt} = \frac{1}{d(d-1)} \sum_{i=1}^{s} \sum_{j \neq i} \left(\frac{S_i^\top S_j}{||S_i|| ||S_j||} \right)^2 \tag{3.11}$$

Discriminative Losses

The main goal of a discremenator is to be able to distinguish fakes samples from real samples and to be able to predict sample labels. To ensure that this is achieved the following conditions have to be enforced:

1. If two nodes do not belong to the same label, then they do not belong to the same cluster. This can be achieved with simple multi-class cross entropy loss which is described as showed in equation 3.12.

$$loss_{xl} = -\mathbb{E}_{\mathbf{x}_i \in X^L} \log P(y_i | \mathbf{x}_i, y_i < L) \tag{3.12}$$

   Where x_i is a labeled sample that is forced to have a label from the L known labels.

2. For each data view push all samples mapping away from central area to maintain the density gap. This is maintained by the property of the GAN loss function described in 3.7. The GAN loss function is described as showed in following equation 3.13

$$\begin{aligned} loss_{gan} = &- \mathbb{E}_{x_i \in X^U} \log[1 - P(L|x_i)] \\ &- \mathbb{E}_{x_i \sim G(z)} \log P(L|x_i) \end{aligned} \tag{3.13}$$

3. Each sample can have only one label that is mapped into a single cluster. A simple solution to

satisfy this condition is to enforce a Cross entropy loss function. Similar to the loss function used in 3.12 for labeled images we implement the same function for unlabeled (predicted label) images. the final unlabeled cross entropy loss function is described in Equation 3.14.

$$loss_{xu} = -\mathbb{E}_{\mathbf{x}_i \in X^U} \sum_{y=0}^{L-1} P(y|\mathbf{x}_i, y_i < L) \log P(y|\mathbf{x}_i, y_i < L) \tag{3.14}$$

Algorithm 3 Mini-batch stochastic gradient descent training of mvSGAN

Require: Node View features $\{\mathbf{w}_{ki}\}$, Labels y^L, Embedding Algorithm $\mathscr{A}$, batch size m.
  Calculate $\{\mathbf{w}'_{ki}\}$ according to Eq. (***)
  Calculate $\{\mathbf{q}_{ki}\}$ via $\mathscr{A}$
  Pairwise concatenate each Node View features $\{\mathbf{w}'_{vi}\}$ with $\{\mathbf{q}_{vi}\}$ for $X^L \cup X^U$
  for $v \in$ k views **do**
    repeat
      Sample m labeled samples $\{\mathbf{x}^L_{v_1}, ..., \mathbf{x}^L_{v_m}\}$ from X^L_v
      Sample m unlabeled samples $\{\mathbf{x}^U_{v_1}, ..., \mathbf{x}^U_{v_m}\}$ from X^U_v
      Sample m noise samples $\{\mathbf{z}_1, ..., \mathbf{z}_m\}$ from $p_\mathbf{z}(\mathbf{z})$
      Update the classifier by descending gradients of losses

$$\nabla_{\theta_{D_v}} \frac{1}{m} \sum loss_{gan_v} + \lambda_0 loss_{xu_v} + \lambda_1 loss_{xl_v} + loss_{pt_v}$$

      for t steps **do**
        Sample m unlabeled samples $\{\mathbf{x}^U_{v_1}, ..., \mathbf{x}^U_{v_m}\}$ from X^U_v
        Sample m noise samples $\{\mathbf{z}_1, ..., \mathbf{z}_m\}$ from $p_\mathbf{z}(\mathbf{z})$
        Update the generator by descending gradients of losses:

$$\nabla_{\theta_{G_v}} \frac{1}{m} \sum loss_{fm_v} + \lambda_2 loss_{pt_v}$$

        Forward propagate X^v_m to obtain P^v_m
      end for
    until convergence
  end for
  repeat
    Forward propagate $P^1_m, P^2_m, ..., P^k_m$ to the fusion layer;
    Computer the loss (3.15)
    Use the gradient of $L_{\text{fusion}}(\theta)$ to update weights for all fully connected layers weights in Fusion network.
  until $L_{\text{fusion}}(\theta)$ Converges

Table 3.1: Dataset statistics

Data set	UCI_mf [13]	AwA2 [14]	NUS_W [15]	SenseIT [16]	Wiki [17]
View 1	PFC(216)	CH(2688)	BCM(225)	ACS(50)	TEXT(10)
View 2	FOU(76)	HOG(252)	CC(144)	SSM(50)	IMG(128)
View 3	KAR(64)	LSS(2000)	CH(64)	-	-
View 4	MOR(6)	RGS(2000)	EDH(73)	-	-
View 5	PIX(240)	SIFT(2000)	WAV(128)	-	-
View 6	ZER(47)	SURF(2000)	-	-	-
Nodes	2,000	30,475	55,613	39,410	2,866
Classes	10	50	683	3	10

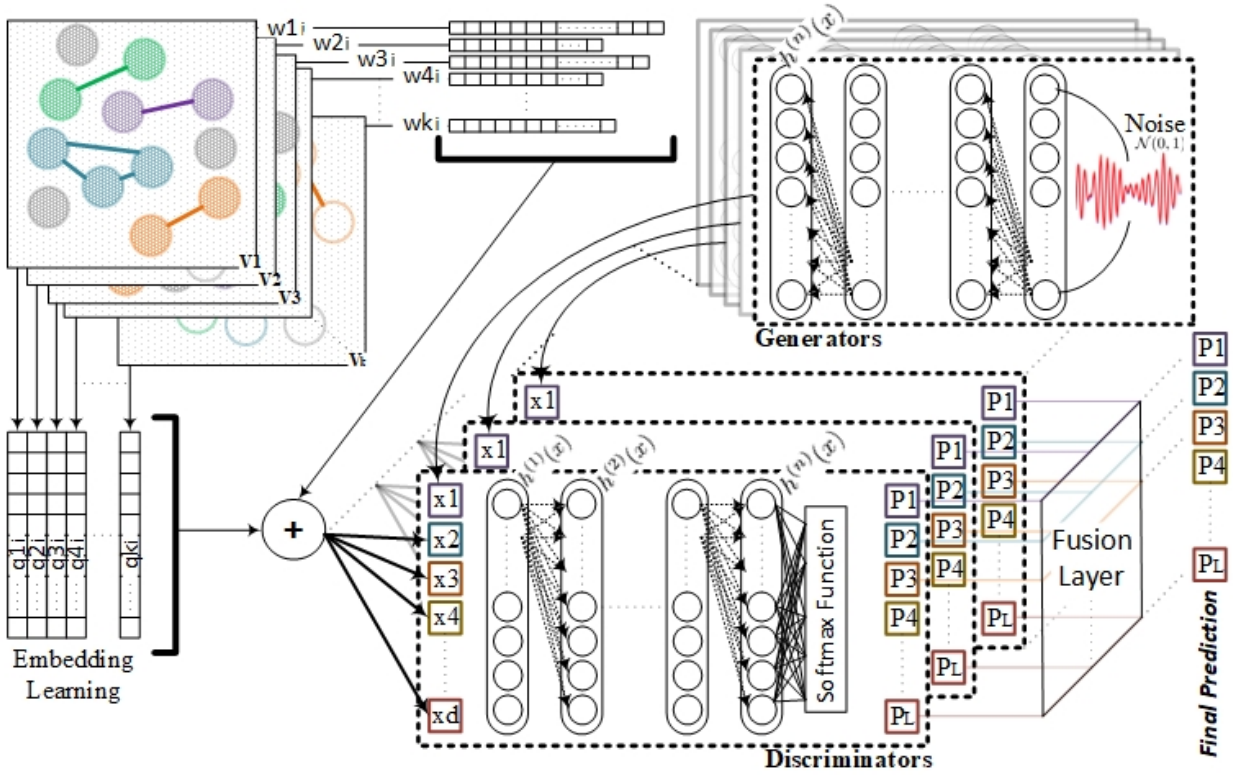

Figure 3.1: mvSGAN model overview.

Fusion Loss

We will implement a voting system between all views probabilistic outputs. The priority of each view is learned through the fusion layer weights. The target is to find the non-linear transformation that will reach the best multi-class cross entropy loss. The loss function is described by equation 3.15 where L is the number of classes, y is the binary indicator (0 or 1) if class label c is correct classification for observation o,and p is the predicted probability observation o is of class

c.

$$L_{\text{fusion}}(\theta) = -\sum_{c=1}^{L} y_{o,c} \log(p_{o,c}) \tag{3.15}$$

3.3.3 Training

Our approach starts by turning nodes in the graph to vectors in feature space. Many methods can be used to accomplish this task [9], [10], [18], [19]. For proof of concept we conduct training using Line method [9]. Neighbor fusion prepossessing step is implemented to accelerate convergence. We alter the method used in [20] by implementing a $k-NN$ classifier trained using the labeled portion of the data set and use that to predict the whole data set edge list. Using the preliminary edge list we calculate $w_i^{'}$ as showed in equation 3.16.

$$\mathbf{w}_i' = \alpha \mathbf{w}_i + \frac{1-\alpha}{k} \sum_{v_j \in Ne(v_i)} \mathbf{w}_j. \tag{3.16}$$

Where $Ne(v_i)$ is the k nearest neighbors of v_i and $k = |Ne(v_i)|$ is the number of neighbors predefined for the Nearest Neighbor classifier.

3.4 Experimental Evaluation

In this section we show the ability of the proposed mvSGAN to achieve better results compared to state of the art methods discussed in the related works section.

3.4.1 Data Sets Description

In the following, we give a brief description of five real-world multi-view benchmark data sets which are used to show the capability of the proposed method.

1. UCI-mf [13]: a numerals ('0'–'9') dataset that consist of 2000 samples and six different views. Each numeral has 10% (200 samples) of the total number of samples.

Table 3.2: Clustering Performance Comparison on Real-World Data Set for a 90% / 10% train / test split. The Best and Second Best are **Boldfaced** and numbered accordingly.

	UCI_mf	AwA2	NUS_W	SenseIT	Wiki
MFMSC	0.4267	0.1823	-	0.3866	0.6866
CoTSC [22]	0.7653	0.2845	-	0.6099	0.7342
MKFC [23]	0.7689	0.4056	0.1462	0.8076	0.4987
MCFCM [24]	0.8176	0.4201	0.2106	0.8320	0.5001
LRRGL [25]	0.8367	0.3722	-	0.7603	0.7609
DS-NMFMVC [26]	0.8905	0.5250	0.4438	0.8158	0.8507
DIMC [27]	0.9302	0.5023	0.4755	0.8330	0.9139
AWP [28]	$\mathbf{0.9687}^{(2)}$	0.4576	-	0.8799	0.9051
MvSN (S) [6]	0.9544	$\mathbf{0.5788}^{(2)}$	$\mathbf{0.5867}^{(2)}$	$\mathbf{0.9027}^{(2)}$	$\mathbf{0.9221}^{(2)}$
mvSGAN	$\mathbf{0.98}^{(1)}$	$\mathbf{0.998}^{(1)}$	$\mathbf{0.993}^{(1)}$	$\mathbf{0.9994}^{(1)}$	$\mathbf{0.9996}^{(1)}$

2. Animal With Attribute (AwA2) [14]: an image dataset set that consist of 37322 samples of 50 animals.

3. NUS Wide [15]: an image dataset that consist of 269648 samples collected from the world wide web.

4. SensIT [21]: a sensor records dataset of approximately 98528 wireless distributed sensor records for three different vehicles in an intelligent transportation system.

5. Wikipedia-Info (Wiki) [17]: a dataset of images and text describing different sections in 11 Wikipedia articles.

We summarize the data sets information in Table (3.1). All methods in this experiments are run 10 times with randomized initialization, and the mean of the results is reported.For each sample x_i, the true label is l_i and the predicted label optioned from our approach is y_i. We compare methods using unsupervised clustering accuracy (ACC) which is defined in equation 3.17.

$$\text{ACC} = \frac{\sum_{i=1}^{N} \delta(l_i, \text{map}(y_i))}{N} \tag{3.17}$$

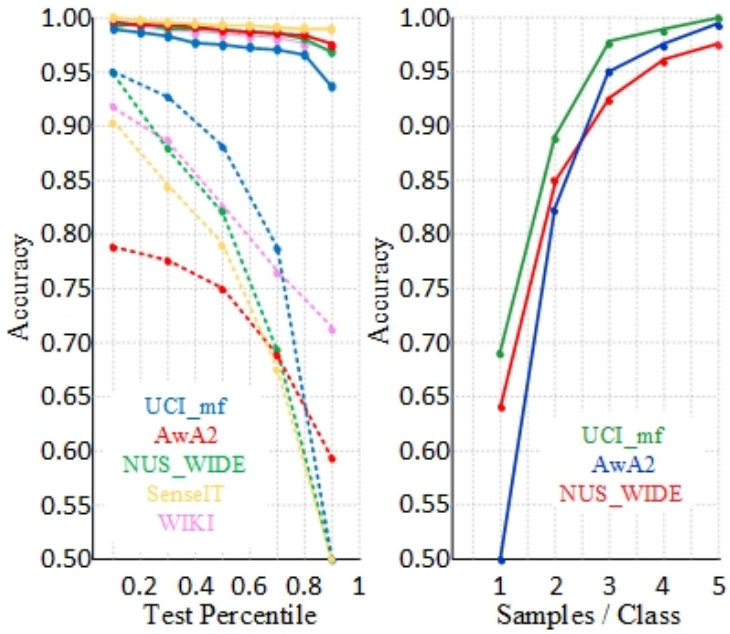

Figure 3.2: GraphSGAN in dashed lines vs mvSGAN in solid lines

3.4.2 Comparison and Discussion of Results

In order to fully test the capabilities of mvSGAN, we compare the accuracy of our implementation of the methods mentioned in Table (3.2). It is demonstrated that our approach mvSGAN that uses GANs to learn the clustering space of samples outperforms all other algorithms compared with including GraphSGAN. It is also demonstrated that mvSGAN performs steadily on multiple different genres of datasets including images, text, and sensor data. While all other popular approaches performance decrease with the increase of number of classes, mvSGAN performs significantly semi-optimal and achieves almost optimal accuracy with 0.05% error as showed in results for NUS_WIDE dataset. The key strength point of mvSGAN is the curse of dimensionality. The higher the dimension of feature view, the better the performance of mvSGAN on those views. Last, The approach is topped with a multi-class cross entropy fusion layer to complete the missing data in views using correct data on other views. This allows mvSGAN to achieve near optimal accuracy and proves the power of GANs in graph theory based image classification.

3.4.3 Scalability Test and Few-Shot Learning

To fully grasp the power of mvSGAN model, we tested the implementation using different training/testing splits. The results in Figure 3.2 are the achieved accuracy of mvSGAN against MvSN apprach 3.2.3 for testing portions of 10% to 90% of the samples with an increment of 10% for all five real world dataset. The clustering performance of mvSGAN is superior due to the sample/class ratio. Unlike spectral clustering, GANs perform much better with low sample/class rate and achieves high accuracy even in low portions of training samples. We test mvSGAN model under few shot learning conditions, we train the model on 1 to 5 samples per class. Our model achieves significantly high accuracy with low sample rates which make it a superior candidate for the task of few shot learning.

3.5 Conclusion

In this paper, we present mvSGAN, a multivew clustering model that utilizes GAN based deep learning approach for multi-class classification. The model is able to reach near optimal clustering prediction for large data sets. The model trains GANs for each view embedding and then fuses the output predictions by assigning weights for each view and finding the best formulation of the views. The algorithm performs significantly when utilized for the task of few shot learning. Experiment results on five benchmark real world data sets show the superiority of our model when compared against state of the art algorithms as demonstrated. Future research include but not limited to extension of GAN training to unified GAN model instead of current GAN per view approach and further enhancements of mvSGAN for single view few-shot learning.

3.6 Acknowledgement

This work was supported in part by Arkansas Research Alliances, National Science Foundation under grant 1946391, and National Institute of Health under grant P20GM121293.

References

[1] H. Salman and J. Zhan, "Similarity metric for millions of unlabeled face images," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 1033–1040.

[2] Z. Wei, M. Xu, L. Geng, H. Liu, and H. Yin, "Adversarial similarity metric learning for kinship verification," *IEEE Access*, vol. 7, pp. 100 029–100 035, 2019.

[3] T. Zhang, Y. Gao, L. Chen, G. Chen, and S. Pu, "Efficient similarity search on quasi-metric graphs," *IEEE Access*, vol. 7, pp. 101 496–101 512, 2019.

[4] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, *Spectralnet: Spectral clustering using deep neural networks*, 2018. arXiv: 1801.01587 [stat.ML].

[5] B. Franci and S. Grammatico, "A game-theoretic approach for generative adversarial networks," *ArXiv*, vol. abs/2003.13637, 2020.

[6] S. Huang, K. Ota, M. Dong, and F. Li, "Multispectralnet: Spectral clustering using deep neural network for multi-view data," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 4, pp. 749–760, 2019.

[7] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ser. ICML'03, Washington, DC, USA: AAAI Press, 2003, pp. 912–919, ISBN: 1-57735-189-4. [Online]. Available: http://dl.acm.org/citation.cfm?id=3041838.3041953.

[8] W. Tang, H. Zhuang, and J. Tang, "Learning to infer social ties in large networks," in *Machine Learning and Knowledge Discovery in Databases*, D. Gunopulos, T. Hofmann, D.

Malerba, and M. Vazirgiannis, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 381–397, ISBN: 978-3-642-23808-6.

[9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line," *Proceedings of the 24th International Conference on World Wide Web - WWW '15*, 2015. DOI: 10.1145/2736277.2741093. [Online]. Available: http://dx.doi.org/10.1145/2736277.2741093.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk," *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, 2014. DOI: 10.1145/2623330.2623732. [Online]. Available: http://dx.doi.org/10.1145/2623330.2623732.

[11] C. N. dos Santos, I. Padhi, P. Dognin, and Y. Mroueh, *Generative feature matching networks*, 2019. [Online]. Available: https://openreview.net/forum?id=Syfz6sC9tQ.

[12] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[13] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[14] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, Sep. 2019, ISSN: 1939-3539. DOI: 10.1109/tpami.2018.2857768. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2018.2857768.

[15] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "Nus-wide: A real-world web image database from national university of singapore," in *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., July 8-10, 2009.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 27:1–27:27, 3 2011, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[17] N. Rasiwasia, J. C. Pereira, E. Coviello, *et al.*, *A new approach to cross-modal multimedia retrieval*.

[18] J. Weston, F. Ratle, and R. Collobert, "Deep learning via semi-supervised embedding," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, Helsinki, Finland: Association for Computing Machinery, 2008, pp. 1168–1175, ISBN: 9781605582054. DOI: 10.1145/1390156.1390303. [Online]. Available: https://doi.org/10.1145/1390156.1390303.

[19] Z. Yang, W. W. Cohen, and R. Salakhutdinov, *Revisiting semi-supervised learning with graph embeddings*, 2016. arXiv: 1603.08861 [cs.LG].

[20] M. Ding, J. Tang, and J. Zhang, "Semi-supervised learning on graphs with generative adversarial nets," *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Oct. 2018. DOI: 10.1145/3269206.3271768. [Online]. Available: http://dx.doi.org/10.1145/3269206.3271768.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 27:1–27:27, 3 2011, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[22] L. Feng, L. Cai, Y. Liu, and S. Liu, "Multi-view spectral clustering via robust local subspace learning," *Soft Comput.*, vol. 21, no. 8, pp. 1937–1948, Apr. 2017, ISSN: 1432-7643. DOI: 10.1007/s00500-016-2120-3. [Online]. Available: https://doi.org/10.1007/s00500-016-2120-3.

[23] H. Huang, Y. Chuang, and C. Chen, "Multiple kernel fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, 2012.

[24] Y. Jiang, F. Chung, S. Wang, Z. Deng, J. Wang, and P. Qian, "Collaborative fuzzy clustering from multiple weighted views," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 688–701, 2015.

[25] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, "Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering," *CoRR*, vol. abs/1608.05560, 2016. arXiv: 1608.05560. [Online]. Available: http://arxiv.org/abs/1608.05560.

[26] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," 2017. [Online]. Available: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14647.

[27] L. Zhao, Z. Chen, Y. Yang, Z. J. Wang, and V. C. Leung, "Incomplete multi-view clustering via deep semantic mapping," *Neurocomputing*, vol. 275, pp. 1053–1062, Jan. 2018. DOI: 10.1016/j.neucom.2017.07.016.

[28] F. Nie, L. Tian, and X. Li, "Multiview clustering via adaptively weighted procrustes," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '18, London, United Kingdom: Association for Computing Machinery, 2018, pp. 2022–2030, ISBN: 9781450355520. DOI: 10.1145/3219819.3220049. [Online]. Available: https://doi.org/10.1145/3219819.3220049.

4 Chapter 4

WaveNets: Wavelet Channel Attention Networks

Hadi Salman, Caleb Parks, Shi Yin Hong, Justin Zhan

Abstract– Channel Attention reigns supreme as an effective technique in the field of computer vision. However, the proposed channel attention by SENet suffers from information loss in feature learning caused by the use of Global Average Pooling (GAP) to represent channels as scalars. Thus, designing effective channel attention mechanisms requires finding a solution to enhance features preservation in modeling channel inter-dependencies. In this work, we utilize Wavelet transform compression as a solution to the channel representation problem. We first test wavelet transform as a standalone channel compression method. We prove that global average pooling is equivalent to the recursive approximate Haar wavelet transform. With this proof, we generalize channel attention using Wavelet compression and name it WaveNet. Implementation of our method can be embedded within existing channel attention methods with a couple of lines of code. We test our proposed method using ImageNet dataset for image classification task. Our method outperforms the baseline SENet-34, and SOTA FcaNet-34.

4.1 Introduction

In deep convolutional neural networks (CNNs), effective feature learning often relies upon the success of attention mechanisms in selectively capturing and preserving relevant important details from input [1]. In tasks such as image classification, attention mechanisms involve redistributing the weights of input feature maps to achieve better classification accuracy [2], [3]. Major attention mechanisms used in CNNs consist of channel attention, spatial attention, branch attention, and temporal attention. [4], [5]. Particularly, the computer vision domain conventionally adopts the channel attention (CA). Introduced by the SeNet [6], CA offers a relatively computationally efficient selection of important channels by generating scalar channel weights, whereby

channel-wise computations are performed on features derived from the global average pooling (GAP).

While channel attention is an intuitive technique in capturing salient properties of images, recent studies suggest that CA's use of global average pooling (GAP) in its architecture hinders its performance. GAP is insufficient in retaining sophisticated details and fails to comply with some task-specific model practices [7]. Moreover, GAP's straightforward dimensionality reduction further limits CA's inter-channel dependencies modeling [8]. Our motivation to design WaveNet stems from this need to reassess CA to capture finer details in feature learning. This reassessment should allow CA to redistribute the weights of input feature maps to improve classification accuracy while maintaining CA's computational efficiency.

To address the above limitations, we propose to enhance the feature preservation during downsampling via the discrete wavelet transform (DWT). As a tool in digital signal processing, DWT has various image processing applications in tasks such as image compression, dehazing, classification, denoising, restoration, and watermarking [9] [10] [11] [12] [13]. Essentially, DWT performs pyramidal image decomposition by transforming an image into four sub-bands composed of a lowpass (LL) filter and a bandpass filter with horizontal (LH), vertical (HL), and diagonal (HH) decomposition of the image, respectively [14] [15]. The LL filter corresponds to a downsized version of the original image with lower resolution, and the LH, HL, and HH bandpass filters highlights the input's predominant traits in their associated orientation. The ability of DWT to perform multilevel decomposition on images inspired WaveNet, in which we explore the levels of decomposition of DWT's application in CA.

In this work, we introduce a novel channel attention framework that stems from a mathematical compression technique. In an effort to better represent channel information and express what GAP failed to explore, we propose to utilize Haar DWT for the channel attention mechanism. Along with the Haar channel attention framework, we propose a customized wavelet channel attention framework. In this framework, we use a set of random orthogonal filters to be used in a customized wavelet. The role of those orthogonal filters is to enforce feature preservation and

diversity in the compression task prior to excitation of channel attention.

Our implementation of this enhanced channel attention mechanism achieves the state-of-the-art performance against related channel attention techniques. The main contribution of this work are summarized as follows:

- We view the channel attention from a compression perspective and adopt DWT in the vanilla channel attention for channel information preservation. With the proof, we establish that conventional GAP is the recurrent Approximate Component of Discrete Haar Wavelet Transform. Then, we generalize the channel attention from the frequency basis and propose our method, termed as WaveNet.

- Motivated by the success of the Discrete Haar Wavelet Transform in WaveNet, we propose WaveNet-C, a custom orthogonal linearly independent filters wavelet to enforce diversity in the compression task for Channel Attention.

- We propose a filter selection criteria along with a parameter reduction technique to fulfill WaveNet-C.

- We conduct extensive experimental results which support that the presented method achieves the state-of-the-art results on ImageNet comparable computational cost to SENet.

4.2 Related Work

Visual Attention in CNNs The active field of research in attention mechanisms has varieties of vision applications across various domains [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27]. Early interest in visual attention is fostered by the highway network [28], which introduced a gating mechanism that enhances the flow of feature information in a deep neural network. ResNet's [29] success with deep CNNs via the use of skips connections in residual blocks further set the foundation for using attention in creating the next state-of-art model. Soon, the proposal of SENet [6] presents the channel attention in an efficient squeeze and excitation architecture, fueling a wave

of studies aiming to improve the channel attention performance. Notably, DANet [30] integrates a position attention module with channel attention to model long-range contextual dependencies. Building upon NLNet [31] and SENet, GCNet [32] proposes GC blocks to capture channel-wise interdependencies while emphasizing long-range global context modeling. [33] introduces the triplet attention, modeling spatial attention and channel attention with efficient parameters and no dimensionality reduction. HA-CNN [34] assesses joint attention selection, which combines hard regional and soft spatial attention with channel attention. Besides utilizing spatial attention, CBAM [35] applies global max-pooling in channel attention to counter GAP's limits. ECA-Net [36] remodels the channel attention architecture to capture cross-channel interaction without unnecessary dimensionality reduction. TSE [37] disregards GAP's global spatial context in SENet to streamline the SE network usage with AI accelerators. FcaNet [38] adds a multi-spectral component to channel attention from a frequency analysis perspective that explains the relationship between GAP and discrete cosine transform.

Wavelet Transforms in Image Processing Wavelet transforms attract growing interest in deep learning-based image processing applications. Early associated works tend to neglect the use of attention mechanisms and range from image super-resolution [39], [40], classification [41] [42], inpainting [43], demoiréing [44], and restoration [12]. Recently, some works propose to integrate attention mechanisms and Wavelet transforms. AWNet [45] integrates non-local attention with DWT to achieve better results for image signal processor (ISP) with smartphone images. [46] proposes WAEN, which composes of an attention embedding network and a wavelet embedding network to enhance video super-resolution. The soft attention-based model proposed in [47] applies DWT to improve face recognition of morphed images. [48] presents a framework for detecting surface defects of glass bottles that fuses the Wavelet transform into their visual attention model. [49] details a single image deraining framework based on a fusion network with DWT and its inverse into its attention module. Different than most previous works that use wavelet transform with an attention mechanism for a specific domain-based application, our WaveNet seeks to incorporate

Wavelet transform into the underlying architecture of CA to improve the attention mechanism at its most fundamental level.

4.3 Method

We start this section by formulating the Discrete Wavelet Transform (DWT) and Channel Attention (CA). We then look into more details over the derivation of our Interdependent channel attention. Together with the interdependent channel attention model, we explore a diversification strategy for custom wavelet transform.

4.3.1 Discrete Wavelet Transform (DWT) and Channel Attention (CA)

In this section, we first go in-depth over the mathematical derivation of DWT. Then, we elaborate on explaining the channel attention mechanism.

DWT using Multiplication Given scale weights H and shift weights G describing wavelet w, the wavelet transform of (1D) input X is:

$$X_{J=1}^{output} = W \times X = \begin{bmatrix} H \\ \text{—} \\ G \end{bmatrix} \times X. \tag{4.1}$$

Where

$$W = \begin{bmatrix} h & 0 & \dots & 0 \\ 0 & h & \ddots & 0 \\ \vdots & \ddots & h & 0 \\ 0 & \dots & 0 & h \\ g & 0 & \dots & 0 \\ 0 & g & \ddots & 0 \\ \vdots & \ddots & g & 0 \\ 0 & \dots & 0 & g \end{bmatrix}_{n \times n} \tag{4.2}$$

The (2D) DWT can be described using the (1D) DWT by applying the procedure to columns first then repeating the process to the rows of the output. The first level DWT for input X can be modeled as follows:

$$\begin{aligned} X_{J=1}^{output} &= DWT(X) = W \times X \times W^T \\ &= \begin{bmatrix} H \\ \hline G \end{bmatrix} X \left[H^T \;\middle|\; G^T \right] \end{aligned} \tag{4.3}$$

$$X_{output} = \begin{bmatrix} \mathscr{A} & \mathscr{V} \\ \mathscr{H} & \mathscr{D} \end{bmatrix} = \left[\begin{array}{c|c} HXH^T & HXG^T \\ \hline GXH^T & GXG^T \end{array}\right] \tag{4.4}$$

Where $\mathscr{A}$ is the Approximation of X, $\mathscr{V}$ is the Vertical difference of X, $\mathscr{H}$ is the Horizontal difference of X, and $\mathscr{D}$ is the Diagonal difference of X. For an image with the size of $(n \times n)$, the extracted features size is $(n/2 \times n/2)$. The wavelet transform level is the number of the times of wavelet feature extraction. At J^{th} level, the extracted feature size is $(n/2^J \times n/2^J)$.

DWT using Convolution Given decoding high pass and low pass filters H, L respectively, we use convolution with correlations of the form :

$$Y_{k,l} = \sum \psi_{ij} X_{i+k,j+l}. \tag{4.5}$$

We assemble the encoding filters by stacking the Low-Low, horizontal, vertical, diagonal filters. For a d value filter where $d \in \{2,3,4,5\}$, the filter bank per channel has the dimensions of size $(4,d,d)$. Haar has $d = 2$ value filter. For input of size $(N \times C \times H \times W)$ the filter bank size for convolution is $(4 \times C, C, d, d)$ with 2 as the stride and no padding. The convolution output is of size $(N, C, 4, H/2, W/2)$ where $\mathscr{A}, \mathscr{V}, \mathscr{H}, \mathscr{D}$ are stacked on 3^{rd} dimension.

Channel Attention Convolution Neural Networks rely heavily on channel attention mechanisms. The idea is to re-calibrate the channel weights based on relative importance to the general task. Suppose that $X \in \mathbb{R}^{C \times H \times W}$ is an instance of a deep image feature, C is the channel count, H is the feature height, and W is the feature width. As discussed in Sec. 4.1, the channel attention process aims to summarize the channel content into a scalar value. Hence the channel attention mechanism described initially by SENet [50] can be written as:

$$\begin{aligned} att &= excite(squeeze(X))) \\ &= sigmoid(fc(GAP(X))) \end{aligned} \tag{4.6}$$

where $att \in \mathbb{R}^C$ is attention vector, *sigmoid* is Sigmoid function, fc is a mapping function such as a fully connected layer or an one-dimensional convolution, and squeeze (GAP): $\mathbb{R}^{C \times H \times W} \mapsto \mathbb{R}^{C \times 1 \times 1}$ is a compression method. After acquiring the attention vector of all C channels, all channels of input X are scaled by their corresponding importance value:

$$\widetilde{X}_{N,C,W,H} = att_{N,C,1,1} \cdot X_{N,C,W,H} \tag{4.7}$$

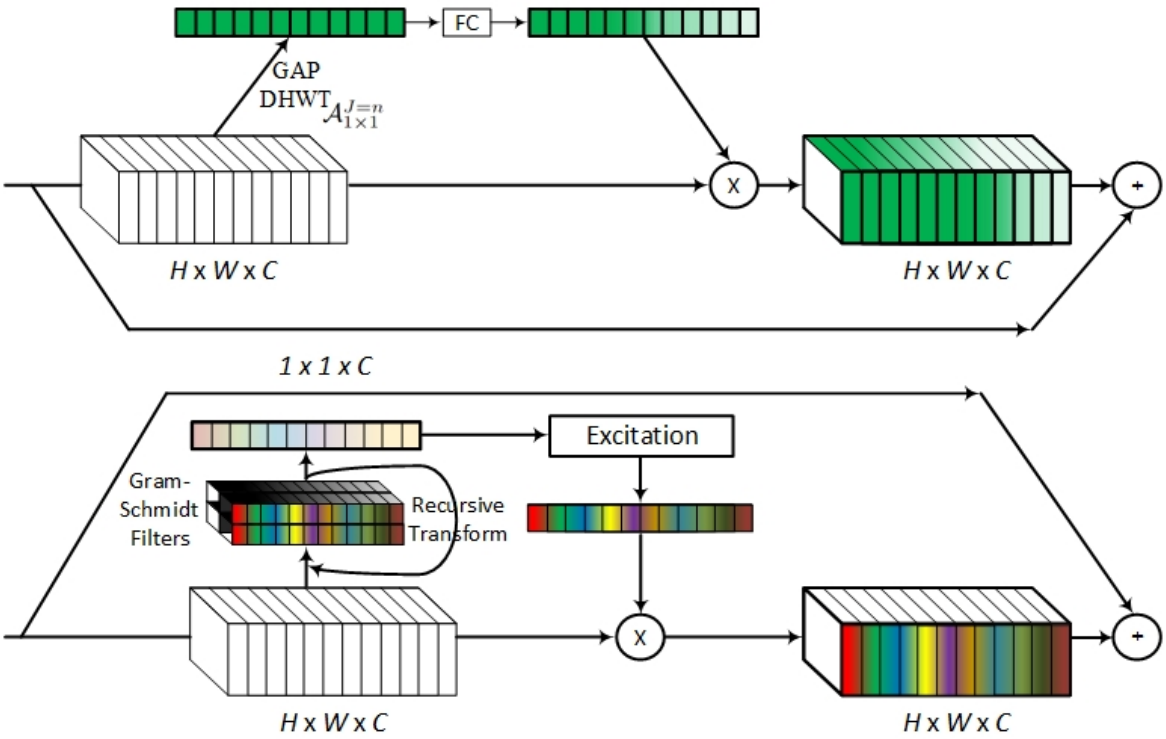

Figure 4.1: Illustration of WaveNet channel attention

where att, X, and $\widetilde{X}$ are the input, attention vector and attention mechanism output.

Typically, global average pooling is used as the compression method [6], [36]. Other popular compression methods are global max pooling [35] and global standard deviation pooling [51].

4.3.2 Interdependent Channel Attention

In this section, we start by highlighting weaknesses of the current channel attention mechanisms. Based on the theoretical analysis, we then discuss our proposed design to overcome those weaknesses.

Wavelet Channel Attention As discussed in Sec. 4.3.1, DWT extracts four main features of an image. With the proof, We demonstrate that GAP is equivalent to the recurrent approximation of the input image when Haar wavelet transform is used.

Theorem 1. *For an image X with the size of $H \times W$, GAP is an exceptional case of 2D DWT with*

result proportional to the $\log_2(\max(H,W))$ *level approximation using 2D Discrete Haar Wavelet Transform (DHWT).*

Proof. The proof is divided into two transforms. The first transform is applied to the input image X to get a padded version with equivalent global average pooling. If the image X isn't divisible by 2 in both dimensions, we pad X to get an image $A = P(X)$ with $GAP(X) = GAP(A)$. If the input image is already divisible by 2 in both dimensions, we define P to be the identity function.

Next we get $GAP(B) = GAP(P(A)) = GAP(X)$. You can repeat this argument until B is a 1x1 and $GAP(B) = b$. To do so, we introduce the second transform T. If $B = T(A)$, for T being the transform, we have

$$B_{i,j} = (A_{2i,2j} + A_{2i+1,2j} + A_{2i,2j+1} + A_{2i+1,2j+1})/4. \tag{4.8}$$

From this, it follows that

$$\sum_{i=1,j=1}^{I,J} B_{i,j} = 1/4 * \sum_{i=1,j=1}^{2*I,2*J} A_{i,j} \tag{4.9}$$

which implies that $GAP(B) = GAP(A)$. Since $GAP(A) = a$ if a is a 1x1 matrix, the proof is complete by induction. □

Orthogonal Linearly Independent Channel Attention Module Theoretical analysis and Theorem 1 support that GAP in the channel attention mechanism only uses the average approximation feature while a diverse variety of potential features are discarded. However, the discarded features may also encode the useful information patterns in representing the channels and should be taken into consideration in the compression phase. To mathematically derive a more diverse and meaningful compression method of channel information, we propose to generalize GAP to more wavelet filters and compress more information with multiple different wavelets.

ResNet has two main blocks, Basic Block and Bottleneck Block. Basic has 4 channel sizes 64, 128, 256, and 512. Bottleneck has 6 channel sizes 64, 128, 256, 512, 1024, and 2048. In case of basic Block, we initialize 4 random interdependent orthogonal filters of same size as channels and

we train the network on those filters. For the Bottleneck, we use the same filters for the channels sizes that are shared with the Basic Block. For the channels 1024 and 2048, we split the channels to chunks of size 512 and we initialize those extra 4 filters of size 512 with new random orthogonal weights to enforce catching more diverse information during the compression phase.

The input X is passed through a separate orthogonal linearly independent wavelet compression module to represent diverse interdependent channel information. In this way, we express the basic compression (C_B) as follows:

$$C_B(X) = \mathrm{DWT}_J(X), \tag{4.10}$$

in which the recursive wavelet level J = $\log_2\{H\}$. $X \in \mathbb{R}^{C\times H\times W}$ is the input feature, and $C(X) \in \mathbb{R}^C$ is the C-dimensional vector post compression. Similarly, bottleneck compression (C_{BN}) is described as follows:

$$C(X)_{BN} = \begin{cases} C_B(X) & C = 64, 128, 256, 512 \\ CAT(C_B(X_{512})) & C = 1024, 2048 \end{cases} \tag{4.11}$$

where CAT is the concatenation function along the channel dimension and X_{512} is the split of the input X of size 512 along the channel dimension.

The final orthogonal interdependent channel attention can be expressed as:

$$Attention(X) = sigmoid(fc(C(X))). \tag{4.12}$$

From Eqs. 4.10, 4.11 and 4.12, it is demonistrated that out model performs a set of Wavelet transforms and extracts channel diverse compression representations of channel information. By incorporating those extra information in the final description we notice a major improvement in the channel representation. Fig. 4.1 illustrates the overall concept of our method.

Table 4.1: Results of the image the classification task on ImageNet over different methods. Besides the AANet, which had no official code implementattion, all methods' results are reproduced and trained with the same training setting.

Method	Years	Parameters	FLOPS	Top-1 acc	Top-5 acc
ResNet [29]	CVPR16	21.80 M	3.68 G	74.58	92.05
SENet [6]	CVPR18	21.95 M	3.68 G	74.83	92.23
ECANet [36]	CVPR20	21.80 M	3.68 G	74.65	92.21
FcaNet-LF	ICCV21	21.95 M	3.68 G	74.95	92.16
FcaNet-TS	ICCV21	21.95 M	3.68 G	75.02	92.07
FcaNet-NAS	ICCV21	21.95 M	3.68 G	74.97	92.34
WaveNet-C	BigData22	21.95 M	3.68 G	**75.06**	92.376

Wavelet Filter Choice One important decision for the network is to pick the wavelet to perform on a specific channel. Our baseline network named Wavenet perform Haar approximation on all channels and achieve SENet results. In order to fulfill the orthogonal interdependent channel attention, we propose Wavenet-C. We discuss more about those networks in the following subsections.

WaveNet means WaveNet weights the components of wavelet compression within each step of the deep wavelet compression. Its main idea is to improve the compression by including the vertical, horizontal, and diagonal components. First, the network determines the importance of each frequency component. Then, it investigates the effect of adding those frequency components together through the recurrence process.

WaveNet-C means WaveNet with selective wavelet filters. We use the convolution based wavelet transform and we assign orthogonal independent filters for channel compression. We do so by randomly initializing the filters then applying the gram-schmidt process to orthogonality those filters thus forcing the network to diversify the information compressed by each channel therefore achieving better classification in general.

4.4 Experiments

In this section, we began by describing the experimental details of our implementation. Then, we discuss the technique of information compression in our framework, complexity, and code implementation. Lastly, we discuss the accuracy of our method on image classification, object detection, and instance segmentation tasks.

4.4.1 Implementation Details

We utilize ResNet-34, as backbone model to evaluate the proposed WaveNet on ImageNet [52]. We comply with data augmentation and hyper-parameter settings in [29] and [53]. Specifically, with random horizontal flipping, the input images are cropped randomly to 256×256. To do so, we modify ResNet architecture to allow the input size to be 256 instead of 224. During training, the SGD optimizer is set with a momentum of 0.9. The learning rate is 0.2, the weight decay is 1e-4, and the batch size is 256 per GPU. All models are trained within 100 epochs using Cosine Annealing Warm Restarts learning schedule and label smoothing. To foster convergence, for every 10 epochs, the learning rate scales by 10% of the previous learning rate. We further adopt the Nvidia APEX mixed precision training toolkit and Nvidia DALI library for fast data loaders for training efficiency.

All models are implemented in PyTorch [54] and tested on two Nvidia Quadro RTX 8000 GPUs.

4.4.2 Discussion

How the Orthogonal Linearly Independent filters compresses and embeds more information In Sec. 4.3.2, we prove that solely adopting the vanilla GAP in the channel attention discards information from all filters except the Haar filter, i.e., GAP. Therefore, designing the filters to be orthogonal and linearly independent using the Gram-Schmidt method would force the network to diversify the information extracted in the channel attention compression phase.

We also provide a theoretical basis to show that more information could be embedded. By nature, deep networks are redundant [55], [56]. If two channels contain redundant information, then the application of GAP on these channels are likely to return repetitive information. On the other hand, our multi-spectral framework extracts less superfluous information from redundant channels since the inherent diverging frequency components contain different information. Thus, our multi-spectral framework can embed more unique salient information in the channel attention mechanism.

Complexity analysis We analyze the complexity of our framework through the number of parameters and the computational cost. Our method does not impose no extra parameters compared with the baseline SENet that introduced the vanilla channel attention since the filters of 2D DWT are pre-computed constant. The negligible increase in the computational cost is also similar to computational cost of SENet. With ResNet-34 backbone, the relative computational cost increases of our method is 0.05% compared with SENet, respectively. More results can be found in Table 4.1.

A Few lines of code change Another strength of the proposed wavelet attention framework is that it can be integrated into existing diverse variants of channel attention implementations. The major distinction between our method and SENet is the adoption of different channel compression method (multi-spectral 2D DWT vs. GAP). As discussed in Sec. 4.3.1 and Eq. 4.5, 2D DWT can be viewed as a constant filter convolution of inputs. It can be simply implemented via a Conv2D layer. Accordingly, arbitrary channel attention methods can adopt our framework easily.

4.5 Conclusion

In this paper, we proposed the WaveNet, an efficient, flexible framework for improving channel attention's power in capturing salient features that can easily incorporate into existing channel attention-based models. Theoretically, we prove that the conventional GAP is the recurrent approximation component of the DHWT that discards all channel information in all filters except the Haar filter. Hence WaveNet tackles channel attention as a compression problem and introduces DWT to preserve more unaccounted channel-wise features under GAP. We further introduce WaveNet-C, a custom orthogonal linearly independent wavelet to best fit the compression task for channel attention, and effective wavelet filter selection criteria and parameter reduction techniques. Empirically, our method persistently improves the performance of channel attention mechanism in ImageNet classification task without raising significant parameters and computation costs relative to existing frameworks. Our future works include extending our method for bigger ResNet networks like ResNet-50, ResNet-101; introducing other tasks and datasets like segmen-

tation and object detection on COCO dataset; and incorporating delayed learning for the wavelet filters to further improve our method accuracy.

4.6 Acknowledgment

This work was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award P20GM139768, and the Arkansas Integrative Metabolic Research Center at the University of Arkansas. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

[1] "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.03.091.

[2] J. Sun, J. Jiang, and Y. Liu, "An introductory survey on attention mechanisms in computer vision problems," in *2020 6th International Conference on Big Data and Information Analytics (BigDIA)*, 2020, pp. 295–300. DOI: 10.1109/BigDIA51454.2020.00054.

[3] H. Salman and J. Zhan, "Similarity metric for millions of unlabeled face images," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 1033–1040. DOI: 10.1109/CCWC47524.2020.9031220.

[4] M. Guo, T. Xu, J. Liu, *et al.*, "Attention mechanisms in computer vision: A survey," *CoRR*, vol. abs/2111.07624, 2021. arXiv: 2111.07624. [Online]. Available: https://arxiv.org/abs/2111.07624.

[5] H. Salman and J. Zhan, "Semi-supervised learning and feature fusion for multi-view data clustering," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 645–650. DOI: 10.1109/BigData50022.2020.9378412.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," 2018, pp. 7132–7141.

[7] Z. Yang, L. Zhu, Y. Wu, and Y. Yang, "Gated channel transformation for visual recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 791–11 800. DOI: 10.1109/CVPR42600.2020.01181.

[8] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *2020 IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), 2020, pp. 11 531–11 539. DOI: 10.1109/CVPR42600.2020.01155.

[9] H. Chen, X. He, L. Qing, S. Xiong, and T. Q. Nguyen, "Dpw-sdnet: Dual pixel-wavelet domain deep cnns for soft decoding of jpeg-compressed images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 824–82 409. DOI: 10.1109/CVPRW.2018.00114.

[10] Q. Li, L. Shen, S. Guo, and Z. Lai, "Wavecnet: Wavelet integrated cnns to suppress aliasing effect for noise-robust image classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 7074–7089, 2021. DOI: 10.1109/TIP.2021.3101395.

[11] M. Fu, H. Liu, Y. Yu, J. Chen, and K. Wang, "Dw-gan: A discrete wavelet transform gan for nonhomogeneous dehazing," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 203–212. DOI: 10.1109/CVPRW53098.2021.00029.

[12] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-cnn for image restoration," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2018.

[13] S. Kushlev and R. P. Mironov, "Analysis for watermark in medical image using watermarking with wavelet transform and dct," in *2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, 2020, pp. 185–188. DOI: 10.1109/ICEST49890.2020.9232700.

[14] M. J. Shensa *et al.*, "The discrete wavelet transform: Wedding the a trous and mallat algorithms," *IEEE Transactions on signal processing*, vol. 40, no. 10, pp. 2464–2482, 1992.

[15] G. Othman and D. Q. Zeebaree, "The applications of discrete wavelet transform in image processing: A review," *Journal of Soft Computing and Data Mining*, vol. 1, no. 2, pp. 31–43, Dec. 2020. [Online]. Available: https://publisher.uthm.edu.my/ojs/index.php/jscdm%20article/view/7215.

[16] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided cnn for image denoising," *Neural Networks*, vol. 124, pp. 117–129, 2020, ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.12.024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608019304241.

[17] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using cnn with attention mechanism," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2439–2450, 2019. DOI: 10.1109/TIP.2018.2886767.

[18] L. Li, M. Xu, X. Wang, L. Jiang, and H. Liu, "Attention based glaucoma detection: A large-scale database and cnn model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[19] M. Wu, D. Huang, Y. Guo, and Y. Wang, "Distraction-aware feature learning for human attribute recognition via coarse-to-fine attention mechanism," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12 394–12 401, Apr. 2020. DOI: 10.1609/aaai.v34i07.6925. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6925.

[20] B. Li, Z. Liu, S. Gao, J.-N. Hwang, J. Sun, and Z. Wang, "Cspa-dn: Channel and spatial attention dense network for fusing pet and mri images," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 8188–8195. DOI: 10.1109/ICPR48806.2021.9412543.

[21] Z. Wang, L. Liu, and F. Li, "Taan: Task-aware attention network for few-shot classification," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 9130–9136. DOI: 10.1109/ICPR48806.2021.9411967.

[22] S.-B. Chen, Q.-S. Wei, W.-Z. Wang, J. Tang, B. Luo, and Z.-Y. Wang, "Remote sensing scene classification via multi-branch local attention network," *IEEE Transactions on Image Processing*, vol. 31, pp. 99–109, 2022. DOI: 10.1109/TIP.2021.3127851.

[23] H. Song, Y. Song, and Y. Zhang, "Sca net: Sparse channel attention module for action recognition," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 1189–1196. DOI: 10.1109/ICPR48806.2021.9413102.

[24] Y. Ding, Z. Ma, S. Wen, *et al.*, "Ap-cnn: Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 2826–2836, 2021. DOI: 10.1109/TIP.2021.3055617.

[25] X. Cun and C.-M. Pun, "Improving the harmony of the composite image by spatial-separated attention module," *IEEE Transactions on Image Processing*, vol. 29, pp. 4759–4771, 2020. DOI: 10.1109/TIP.2020.2975979.

[26] S. Li, B. Xie, Q. Lin, C. H. Liu, G. Huang, and G. Wang, "Generalized domain conditioned adaptation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. DOI: 10.1109/TPAMI.2021.3062644.

[27] X. Xue, S.-i. Kamata, and D. Luo, "Skin lesion classification using weakly-supervised fine-grained method," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 9083–9090. DOI: 10.1109/ICPR48806.2021.9412042.

[28] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016, pp. 770–778.

[30] J. Fu, J. Liu, H. Tian, *et al.*, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[31] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[32] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct. 2019.

[33] H. Peng, X. Chen, and J. Zhao, "Residual pixel attention network for spectral reconstruction from rgb images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2020.

[34] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2285–2294.

[35] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," 2018, pp. 3–19.

[36] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," 2020, pp. 11 534–11 542.

[37] N. Vosco, A. Shenkler, and M. Grobman, "Tiled squeeze-and-excite: Channel attention with local spatial context," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct. 2021, pp. 345–353.

[38] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 783–792.

[39] W. Ma, Z. Pan, J. Guo, and B. Lei, "Achieving super-resolution remote sensing images via the wavelet transform combined with the recursive res-net," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3512–3527, 2019. DOI: 10.1109/TGRS.2018.2885506.

[40] B. Lowe, H. Salman, and J. Zhan, *Ghm wavelet transform for deep image super resolution*, 2022. DOI: 10.48550/ARXIV.2204.07862. [Online]. Available: https://arxiv.org/abs/2204.07862.

[41] Q. Li, L. Shen, S. Guo, and Z. Lai, "Wavelet integrated cnns for noise-robust image classification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7243–7252. DOI: 10.1109/CVPR42600.2020.00727.

[42] D. D. N. D. Silva, H. W. M. K. Vithanage, K. S. D. Fernando, and I. T. S. Piyatilake, "Multi-path learnable wavelet neural network for image classification," in *Twelfth International Conference on Machine Vision (ICMV 2019)*, W. Osten and D. P. Nikolaev, Eds., International Society for Optics and Photonics, vol. 11433, SPIE, 2020, pp. 459–467. DOI: 10.1117/12.2556535. [Online]. Available: https://doi.org/10.1117/12.2556535.

[43] Y. Yu, F. Zhan, S. Lu, *et al.*, "Wavefill: A wavelet-based generation network for image inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[44] L. Liu, J. Liu, S. Yuan, *et al.*, "Wavelet-based dual-branch network for image demoiréing," in *European Conference on Computer Vision*, Springer, 2020, pp. 86–102.

[45] L. Dai, X. Liu, C. Li, and J. Chen, "Awnet: Attentive wavelet network for image isp," in *ECCV Workshops*, 2020.

[46] Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Wavelet attention embedding networks for video super-resolution," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 7314–7320. DOI: 10.1109/ICPR48806.2021.9412623.

[47] P. Aghdaie, B. Chaudhary, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Attention aware wavelet-based detection of morphed face images," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*, 2021, pp. 1–8. DOI: 10.1109/IJCB52358.2021.9484398.

[48] X. Zhou, Y. Wang, Q. Zhu, *et al.*, "A surface defect detection framework for glass bottle bottom using visual attention model and wavelet transform," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2189–2201, 2020. DOI: 10.1109/TII.2019.2935153.

[49] H.-H. Yang, C.-H. H. Yang, and Y.-C. F. Wang, "Wavelet channel attention module with a fusion network for single image deraining," in *2020 IEEE International Conference on*

Image Processing (ICIP), 2020, pp. 883–887. DOI: 10.1109/ICIP40778.2020.9190720.

[50] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[51] H. Lee, H.-E. Kim, and H. Nam, "Srm: A style-based recalibration module for convolutional neural networks," 2019, pp. 1854–1862.

[52] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," pp. 211–252, 2015.

[53] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," 2019, pp. 558–567.

[54] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," 2019, pp. 8026–8037.

[55] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," 2017, pp. 1389–1397.

[56] Z. Zhuang, M. Tan, B. Zhuang, *et al.*, "Discrimination-aware channel pruning for deep neural networks," 2018, pp. 875–886.

5 Chapter 5

OrthoNets: Orthogonal Channel Attention Networks

Hadi Salman, Caleb Parks, Matthew Swan

Abstract– Designing an effective channel attention mechanism implores one to find a lossy-compression method allowing for optimal feature representation. Despite recent progress in the area, it remains an open problem. FcaNet, the current state-of-the-art channel attention mechanism, attempted to find such an information-rich compression using Discrete Cosine Transforms (DCTs). One drawback of FcaNet is that there is no natural choice of the DCT frequencies. To circumvent this issue, FcaNet experimented on ImageNet to find optimal frequencies. We hypothesize that the choice of frequency plays only a supporting role and the primary driving force for the effectiveness of their attention filters is the orthogonality of the DCT kernels. To test this hypothesis, we construct an attention mechanism using randomly initialized orthogonal filters. Integrating this mechanism into ResNet, we create OrthoNet. We compare OrthoNet to FcaNet (and other attention mechanisms) on Birds, MS-COCO, and Places356 and show superior performance. On the ImageNet dataset, our method competes with or surpasses the current state-of-the-art. Our results imply that an optimal choice of filter is elusive and generalization can be achieved with a sufficiently large number of orthogonal filters. We further investigate other general principles for implementing channel attention, such as its position in the network and channel groupings.

5.1 Introduction

Deep convolutional neural networks have become the standard tool to accomplish many computer vision tasks such as classification, segmentation, and object detection [1], [2]. Their success is attributed to the ability to extract features related to the underlying task. Higher quality features allow for better outputs in the decision space. Hence, improving the quality of these features has become an area of interest in the machine learning community [3]–[5]. In this paper,

we investigate channel attention mechanisms.

A channel attention mechanism is a module placed throughout the network. Each module takes as input a feature, which has C channels, and outputs a C dimensional attention vector with components in $(0, 1)$. By multiplying these outputs with the input feature vector, the features are adapted towards solving the underlying task.

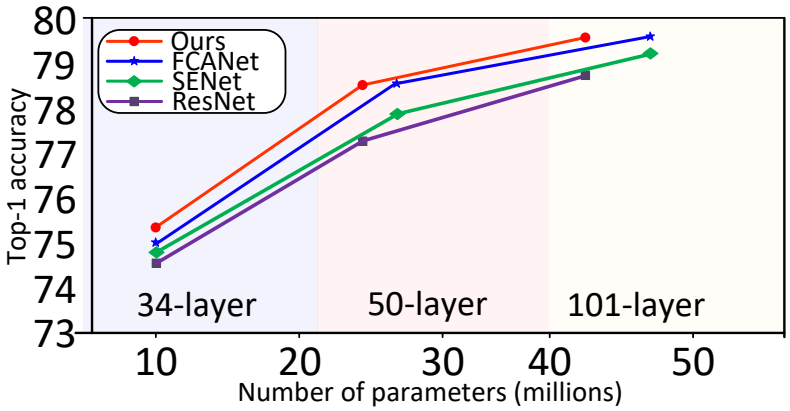

Figure 5.1: ImageNet accuracy comparison.

The first channel attention mechanism, introduced by SENet [6], uses Global Average Pooling (GAP) to compress the spatial dimension of each feature channel into a single scalar. To compute the attention vector, the compressed representation is sent through a Multi-Layer Perceptron (MLP) and then through a sigmoid function. The compression stage is referred to as the squeeze. As pointed out in [7], a major weakness of SENet is using a single method (i.e. GAP) to compress each channel. This weakness was removed by FcaNet [7]. They argue that GAP is discarding essential low-frequency information and that generalizing channel attention in the frequency domain by using DCTs, yields more information. Doing so improves channel attention significantly. Based on imperial results, FcaNet chose which Discrete Cosine Transforms (DCTs) to use, thus becoming the state-of-the-art (SOTA) channel attention mechanism.

We hypothesize that the success of FcaNet has less to do with the low-frequency spectral information and is, in fact, due to the orthogonal nature of the DCT compression mappings. To

test this hypothesis, we construct a channel attention mechanism using random orthonormal filters to compress the spatial information of each feature. Using our attention mechanisms, we produce networks that outperform FcaNet in object detection and image segmentation tasks. Compared to SOTA, our method achieves competitive or superior results on the ImageNet dataset and achieves top performance for attention mechanisms on Birds and Places365 datasets. We further investigate other general principles for implementing channel attention. The main contributions of this work are summarized as follows:

- We propose to diversify the channel squeeze methods in an effort to extract more information using orthogonal filters and name the resulting network OrthoNet. By evaluating OrthoNet, we demonstrate that a key property of quality channel attention is filter diversity.
- We propose to relocate the channel attention module in ResNet-50 and 101 bottleneck blocks. Our location reduces the number of parameters used for our module and improves the network's overall performance.
- Running numerous experiments on Birds, Places365, MS-COCO, and ImageNet datasets; we demonstrate that OrthoNet is the state-of-the-art channel attention mechanism.
- To explore network architecture choices, we investigate channel grouping in the squeeze phase and squeeze filter learning.

The rest of the paper is formatted as follows. In section 2, we review works related to this paper. Section 3 formulates channel attention mechanism and reviews the most important channel attentions. In section 4, we report our results. Section 5 contains discussion on architecture choices and further benefits of this method. Finally, in section 6 we conclude our work and hypothesize why distinct attention filters are key to an attention mechanism.

5.2 Related Work

Deep Convolutional Neural Networks (DCNNs) LeNet [8] and AlexNet [9] served as the starting point for a new era in fast GPU-implementation of CNNs. Since then, researchers have started exploring the potential of adding more convolutional layers while maintaining computational efficiency. To improve the utilization of GPUs, GoogLeNet [10] was proposed based on the Hebbian principle and the intuition of multi-scale processing. Shortly after, VGGNet [11] was introduced and secured first and second place in ImageNet Challenge 2014. Their work began a trend to deepen networks to achieve higher accuracy. As a result, the number of network parameters increased causing difficulty during optimization. To overcome this challenge, ResNet [12] explicitly reformulated the layers as learning residual functions which made the network easier to optimize. Shortly after, many variants of ResNet were introduced to enhance it and overcome problems like vanishing gradients and parameter redundancy [4], [13], [14].

Visual Attention in DCNNs Based on the concept of focus in human behavior, visual attention aims to highlight the important parts of an image. Literature suggests those important parts are chosen because they are significant to distinguish a specific image from others. The current research in visual attention aims to leverage those properties [15]–[24]. The highway network [25] introduced a basic — yet effective — gating mechanism that promotes the flow of information in deep neural networks. One can consider the "transform gate" from the highway network as a type of attention. Based on ResNet backbone, SENet then presented channel attention using the squeeze and excitation architecture, ushering the start of a research wave aiming to improve the channel attention process.

DANet [26] integrated a position attention module with channel attention to model long-range contextual dependencies. NLNet [27] aggregated query-specific global context to each query position in the attention module. Building upon NLNet and SENet, GCNet [28] proposed the GC-block, which aims at capturing channel cross-talk and inter-dependencies while maintaining global context awareness. Triplet attention [29] explored an architecture that includes spatial and chan-

nel attention while maintaining computational efficiency. CBAM [30] applied global max-pooling instead of GAP in SENet as an alternative. GSoPNet [31] introduced global second-order pooling instead of GAP, which is more effective but computationally expensive. ECANet [32] remodeled the channel attention architecture to capture cross-channel interaction without unnecessary dimensionality reduction.

FcaNet [7] added a multi-spectral component to channel attention from a frequency analysis perspective. They explained the relationship between GAP and Discrete Cosine Transform's initial frequency and then used a selection of the remaining frequencies to extract channel information. Finally, WaveNet [33] proposed to use Discrete Wavelet Transform to extract channel information.

Datasets for Visual Tasks Behind every success in deep neural network tasks, there exists a dataset that was curated to guide those networks to generalize. ImageNet [34] is considered the most popular image dataset mainly used for classification; it has more than 14 million hand-annotated images. A very commonly used subset is ImageNet-1000 which has 1000 classes, 1.3 million images for training, and 50 thousand images for validation. Another popular dataset is MS-COCO [35] consisting of approximately 118 thousand annotated training images for object detection, key-points detection, and panoptic segmentation. It has 5 thousand validation images. Places365 [36] is a dataset for scene recognition, that consist of 1.8 million training images from 365 scene categories with 36 thousand validation images. Finally, Birds [37] is a medium size relatively easy classification dataset of different types of birds that are used to evaluate models on low-level features. The dataset consist of 450 species, each with more than 150 samples for training and five samples for validation.

5.3 Method

In this section, we review the general formulation of channel attention mechanisms. With this formulation, we briefly review SENet and FcaNet. Based on these works, we introduce our

method, Orthogonal Channel Attention, to be implemented in OrthoNet.

5.3.1 Channel Attention

Considered a strongly influential attention mechanism, channel attention was first proposed in [6] as an add-on module to be incorporated into any existing architecture. Its goal is to improve overall network performance with negligible computational cost.

A *channel attention block* is a computational unit, built to encapsulate information and highlight relevant features. Suppose $X \in \mathbb{R}^C \times \mathbb{R}^H \times \mathbb{R}^W$ is a feature vector where C is the number of channels with H and W being the height and width. The channel attention computes a vector, $A \in \mathbb{R}^C$, which highlights the most relevant channels of X. The output of the module is $A \odot X$, calculated by

$$(A \odot X)_{c,h,w} = A_c X_{c,h,w}\,. \tag{5.1}$$

Various researchers have proposed variants on Channel Attention that compute the attention vector, A, in different ways.

5.3.2 Squeeze-and-Excitation (SENet)

The Squeeze-and-Excitation method is broken into two stages: a *squeeze* phase followed by an *excitation* stage. The squeeze stage can be considered a lossy-compression method using GAP as shown below:

$$Z_c = \mathbf{F}_{\text{se}}(X)_c = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} X_{c,i,j}\,. \tag{5.2}$$

The excite step maps the compressed descriptor Z to a set of channel weights. This is achieved by

$$\mathbf{E}(Z) = \sigma(W_2\delta(W_1 Z))\,, \tag{5.3}$$

where σ is the sigmoid function, δ is the ReLU activation function, and W_1, W_2 are (learnable) matrix weights. We can formulate channel attention in SENet in the following form:

$$A(X) = \mathbf{E}(\mathbf{F}_{\text{se}}(X)). \tag{5.4}$$

5.3.3 Frequency Channel Attention (FcaNet)

One of the major weaknesses of the channel attention proposed in SENet [6] was the use of only one squeeze method: GAP. The authors were motivated to use GAP to encourage representation of global information in the attention vector. FcaNet [7] proposed an alternative squeeze method based on Discrete Cosine Transform. The DCT for an image of size (H, W) with frequency component (i, j) is given by

$$T_{i,j}(X) = \sum_{h=1}^{H} \sum_{w=1}^{W} v_i(h,H) v_j(w,W) X_{h,w}, \tag{5.5}$$

where

$$v_k(a,A) = \cos\left(\frac{\pi k(a+1/2)}{A}\right). \tag{5.6}$$

They proved that GAP is proportional to the initial DCT frequency, $(0,0)$, and demonstrated that the remaining frequencies need to be represented in the attention vector. They claim that those missing frequencies contain essential information; however, they give no theoretical reasoning to explain why their choice frequencies was optimal. FcaNet attention vector can be computed as follows:

$$A(X) = \mathbf{E}(\mathbf{F}_{\text{fca}}(X)), \tag{5.7}$$

where

$$\mathbf{F}_{\text{fca}}(X)_c = T_{I(c),J(c)}(X_c) \tag{5.8}$$

for some choice of functions I, J on the natural numbers. Based on experimental results, FcaNet chose to break the channel dimension into 16 blocks and chose (I, J) to be constant on each block.

5.3.4 Orthogonal Channel Attention (OrthoNet)

We notice that the DCTs used in FcaNet have a unique, essential, property: they are orthogonal [38]. In this paper, we exploit the benefits of the orthogonal property for channel attention. Roughly, we start by randomly selecting filters of the appropriate dimension, (C, H, W); we then apply Gram-Schmidt process to make those filters orthonormal. The full details for initializing the filters are described in algorithm 4. Denoting these filters by $K \in \mathbb{R}^C \times \mathbb{R}^H \times \mathbb{R}^W$, our squeeze process is given by

$$\mathbf{F}_{\text{ortho}}(X)_c = \sum_{h=1}^{H} \sum_{w=1}^{W} K_{c,h,w} X_{c,h,w} \,. \tag{5.9}$$

As in the other methods, we define our channel attention by

$$A(X) = \mathbf{E}(\mathbf{F}_{\text{ortho}}(X)) \tag{5.10}$$

Algorithm 4 Orthogonal Channel Attention Initialization (Stage Zero)

Ensure: Feature Dimension: (C, H, W).
Require: Output: Kernel $K \in \mathbb{R}^C \times \mathbb{R}^H \times \mathbb{R}^W$.
  if $HW < C$ **then**
    Calculate $n = \text{floor}(C/(HW))$
    Initialize $\mathbf{L}$ as empty list
    for $i \in \{0 \dots n\}$ **do**
      Initialize HW random filters $F_j \in \mathbb{R}^{HW}$
      Run Gram-Schmidt process on $\{F_j\}_{j=1}^{HW}$ to get an orthogonal set $\{F'\}_{j=1}^{HW}$.
      Append $\{F'\}_{j=1}^{HW}$ to List L.
    end for
    Concatenate filters in list L to get kernel $K \in \mathbb{R}^C \times \mathbb{R}^H \times \mathbb{R}^W$.
  else
    Initialize C random filters $F_j \in \mathbb{R}^{HW}$
    Run Gram-Schmidt process on $\{F_j\}_{j=1}^{C}$ to get an orthogonal set $\{F'\}_{j=1}^{C}$.
    Concatenate filters $\{F'\}_{j=1}^{C}$ to get kernel $K \in \mathbb{R}^C \times \mathbb{R}^H \times \mathbb{R}^W$.
  end if

5.4 Experimental Settings and Results

We begin by describing the details of our experiments. We then report the effectiveness of our method on image classification, object detection, and instance segmentation tasks.

5.4.1 Implementation Details

Following [6], [7], [32], we add our proposed attention module to ResNet-34 to construct OrthoNet-34. Based on ResNet-50 and 101, we construct two versions of our network called OrthoNet and OrthoNet-MOD. They differ in the position of the attention module in the ResNet blocks. For further details refer to section 5.5.2.

General Specifications We adopt the Nvidia APEX mixed precision training toolkit and Nvidia DALI library for training efficiency. Our models are implemented in PyTorch [40], and are based on the code released by the authors of FcaNet [7]. The models were tested on two Nvidia Quadro RTX 8000 GPUs.

Classification

Table 5.2: Results of the image the classification task on ImageNet over different methods.

Method	Years	Backbone	Parameters	FLOPS	T1 acc	T5 acc
ResNet [12]	CVPR16	ResNet-101	44.55 M	7.85 G	78.72	94.30
SENet [6]	CVPR18		49.29 M	7.86 G	79.19	94.50
CBAM [30]	ECCV18		49.30 M	7.88 G	78.49	94.31
AANet [39]	ICCV19		45.40 M	8.05 G	78.70	94.40
ECANet [32]	CVPR20		44.55 M	7.86 G	79.09	94.38
FcaNet-LF[7]	ICCV21		49.29 M	7.86 G	79.46	94.60
FcaNet-TS[7]	ICCV21		49.29 M	7.86 G	79.63	94.63
FcaNet-NAS[7]	ICCV21		49.29 M	7.86 G	79.53	94.64
OrthoNet^{+}	CVPR22		49.29 M	7.86 G	79.61	**94.73**
OrthoNet-MOD$^{+,\$}$	CVPR22		44.84 M	7.85 G	**79.69**	94.61

* High computational cost. + Randomly initialized. $ Reduced number of parameters compared to SOTA.

We utilize ImageNet [34], Places365 [36], and Birds [37] datasets to test and evaluate our method. To judge efficiency, we report the number of floating point operations per second

(FLOPs) and the number of frames processed per second (FPS). To demonstrate method effectiveness, we report the top-1 and top-5 accuracies (T1, T5 acc). We use the same data augmentation and hyper-parameter settings found in [7]. Specifically, we apply random horizontal flipping, random cropping, and random aspect ratio. The resulting images are of size 256×256.

During training, the SGD optimizer is set with a momentum of 0.9, the learning rate is 0.2, the weight decay is $1e-4$, and the batch size is 256. All models are trained for 100 epochs using Cosine Annealing Warm Restarts learning schedule and label smoothing. At the beginning of every tenth epoch, the learning rate scales by 10% of the previous learning rate. Doing so fosters convergence, as demonstrated in [7].

Table 5.3: Results of the object detection task on COCO val 2017 over different methods.

Method	Detector	Parameters	FLOPs	AP	AP_{50}	AP_{75}	AP_S
ResNet-50	Faster-RCNN	41.53 M	215.51 G	36.4	58.2	39.2	21.8
SENet		44.02 M	215.63 G	37.7	60.1	40.9	22.9
ECANet		41.53 M	215.63 G	38.0	60.6	40.9	23.4
FcaNet-TS		44.02 M	215.63 G	39.0	61.1	42.3	23.7
OrthoNet-MOD		41.68 M	215.63 G	**39.1**	60.2	42.5	23.4
ResNet101	Faster-RCNN	60.52 M	295.39 G	38.7	60.6	41.9	22.7
SENet		65.24 M	295.58 G	39.6	62.0	43.1	23.7
ECANet		60.52 M	295.58 G	40.3	62.9	44.0	24.5
FcaNet-TS		65.24 M	295.58 G	**41.2**	63.3	44.6	23.8
OrthoNet-MOD		60.84 M	295.58 G	40.8	61.6	44.9	24.7
FcaNet-TS-50*	Mask-RCNN	46.66 M	261.93 G	37.3	57.5	40.5	22.1
OrthoNet-MOD*		44.53 M	261.93 G	**37.8**	57.9	41.1	22.5

* Base configuration used for training. For more details refer to section 5.4.1 and section 5.5.6.

Object Detection and Segmentation

We train and evaluate object detection and segmentation tasks using MS-COCO [35]. We report the average precision (AP) metric and its many variants.

FasterRCNN We use FasterRCNN with OrthoNet-50 and 101 with one frozen stage and learnable BatchNorm. We use the same configuration used by FcaNet [7] based on MMDetection toolbox [41].

MaskRCNN We use MaskRCNN with OrthoNet-50. We have one frozen stage and no learnable BatchNorm. We utilize the base configuration described in MMDetection toolbox [41] of ResNet-50.

Training Configuration We train for 12 epochs. The SGD optimizer is set with a momentum of 0.9. We warm up the model in the first 500 iterations starting with a learning rate of $1e-4$ and growing by 0.001 every 50 iterations. After, we set the learning rate to 0.01 for the first 8 epochs. At epoch 9, the learning rate decays to 0.001 then at epoch 12 it decays to 0.0001. We evaluate both FcaNet and our method using this configuration. For more details, refer to section 5.5.6.

5.4.2 Results

Table 5.4: Results of the scene recognition task on Places365 dataset and classification task on Birds dataset. Our method achieves superior performance compared to FCANet. Both methods are trained with same training settings and using ResNet-50 backbone.

Method	Dataset	T1 acc	T5 acc
FcaNet-TS [7]	Places365 [36]	56.15	**86.19**
OrthoNet-MOD	Places365 [36]	**56.33**	86.18
FcaNet-TS [7]	Birds [37]	97.60	99.47
OrthoNet-MOD	Birds [37]	**97.78**	**99.64**

Classification Results Accuracy results obtained on ImageNet are shown in **??**. The first noticeable feature is OrthoNet's superior performance when using ResNet-34 as the backbone. The result shown in the table is an average over five trials with a standard deviation of 0.12. Even in the worst case, we achieved 74.97 which is comparable to FcaNet. Our results on ResNet-50 and 101 are better than both FcaNet-LF and FcaNet-NAS. The result for OrthoNet-50 is a mean over four trials with standard deviation of 0.07.

The results obtained on ResNet-50 validate our initial hypothesis. If we Consider a linear scale with SENet mapping to zero and FcaNet mapping to one, our method achieves a 0.93. Since SENet has the least orthogonal filters (they are all the same), this demonstrates the importance of orthogonal filters.

We believe that FcaNet slight improvement with ResNet-50 is because they choose particular filters that performed best on their experiments with ImageNet. To test this hypothesis, we evaluated our model on the Places365 and Birds classification datasets. Results are shown in 5.4. We can see that our method outperforms FcaNet-TF on Birds by 0.18% and on Places365 by 0.18%. These results imply that our method can generalize better to different datasets.

Table 5.5: Results of the instance segmentation task on COCO val 2017 over different methods using Mask R-CNN.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M
FcaNet-TS-50	34.0	54.6	36.1	18.5	37.1
OrthoNet-MOD	**34.4**	55.0	36.8	19.0	37.5

Base configuration used for training. For more details refer to section 5.4.1 and section 5.5.6.

Object Detection Results In addition to testing on ImageNet, Places365, and Birds datasets, we evaluate OrthoNet on MS COCO dataset to evaluate its performance on alternative tasks. We use OrthoNet and FPN [42] as the backbone of Faster R-CNN and Mask R-CNN.

As shown in Table 5.3, when OrthoNet-MOD-50 is incorporated into the Faster-RCNN and Mask-RCNN frameworks performance surpasses that of FcaNet. We also achieve 10% less computational cost.

Segmentation Results To further test our method, we evaluate OrthoNet-MOD-50 on instance segmentation task. As demonstrated in table 5.5, our method outperforms FcaNet under the ResNet-50 basic configuration of Mask-RCNN framework. These results verify the effectiveness of our method.

5.5 Discussion

We begin with a possible explanation for the success of OrthoNet. We then detail our explorations into variants of our architecture and conclude by discussing implementation and limitations.

5.5.1 Attention Mechanism Theory

While FcaNet believes that frequency choice for the discrete cosine transform is the primary factor for a successful attention mechanism, our results imply that the key driving force behind successful attention mechanisms is distinct (orthogonal) attention filters. In brief, we believe that the success of FcaNet is mostly due to the orthogonality of the DCT kernels.

Recall that convolutional neural networks contain built-in redundancy [7], [43], [44] (i.e., hidden features with strongly correlated channel slices). In a standard SENet, the squeeze method will extract the same information from these redundant channels. In fact, by appropriately permuting the learnable parameters of a SENet, one can construct another network where the only difference is the order of the channels in the intermediate features. (Such a trick cannot be applied to FcaNet and OrthoNet due to the presence of constant, orthogonal squeeze filters.) The existence of this permutation method implies that SENet cannot, a priori, prescribe meaning to its channels.

When the filters are orthogonal, the information they extract comes from orthogonal subspaces of the feature space ($\mathbb{R}^H \times \mathbb{R}^W$). Hence, they focus distinct characteristics. Since the gradient information flows backward through the network, the convolutional kernels preceding the squeeze can adapt to their unique mapping. By doing so, the network can extract a richer representation for every feature map, which the excitation can then build upon.

To further support our ideas, we train OrthoNet-34 and 50 using random filters — we omit the Gram-Schmidt step — and obtain accuracies of 74.63 and 76.77 respectively. We can clearly see the effect of orthogonal filters on improving the validation accuracy on both networks compared to random filter and GAP (Table **??**).

5.5.2 Attention Module Location

The overall structure of a ResNet-50 or 101 bottleneck block is shown in figure 5.3. To construct OrthoNet, we follow the standard procedure placing the attention module following the 1×1 convolution. Motivated by the below reasoning, we moved the attention in those networks to follow the 3×3 convolution. The resulting network is OrthoNet-MOD.

In OrthoNet-34, the attention is placed on the second 3×3 convolution in each block. We now recall the structure of ResNet 50 and 101: both networks contains many blocks consisting of a 1×1 convolution followed by a 3×3 convolution then another 1×1 convolution (with batch-norms and activations placed between). Since the 1×1 convolutions can only consider inter-channel relationships and lack the ability to capture spatial information, they are mainly used for feature refinement and for channel resizing. Combined with an activation, a 1×1 convolution can be considered as a Convolutional MLP [45]. Since the 3×3 convolutions are the only modules that consider spatial-relationships, they must extract rich spacial information if the network hopes to achieve high accuracies. These motivations and the success of OrthoNet-34 lead to the construction of OrthoNet-MOD.

This modification yields many benefits. First, we reduce the number of parameters used for the attention module by only creating filters of the same size as we did in OrthoNet-34. Exact parameter counts can be found in Table **??**. Second, we improve accuracies over the standard OrthoNet. Third, we place the attention at a more meaningful location, the 3×3 convolution, where features are richer in spacial information.

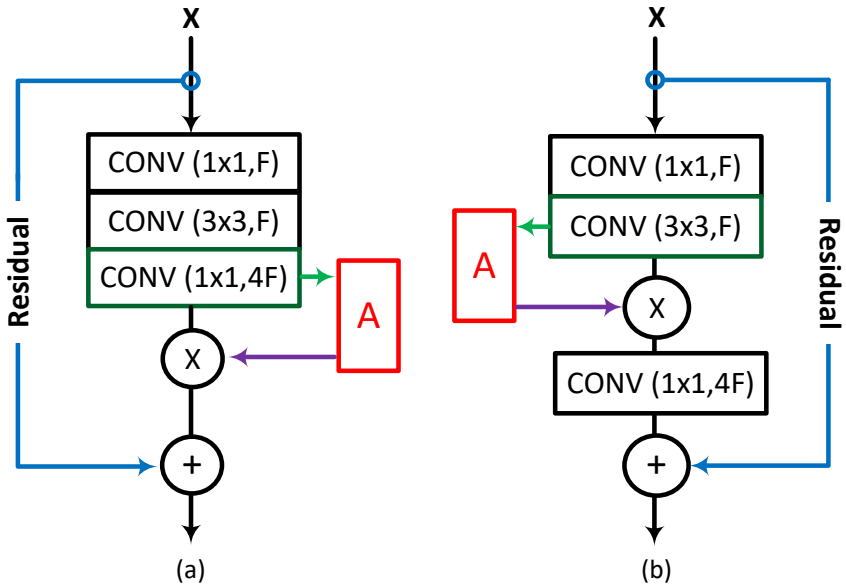

Figure 5.3: (a) OrthoNet block vs (b) OrthoNet-MOD block

5.5.3 Cross-talk Effect on Orthogonal Filters

Our squeeze method can be considered as a convolution with kernel size equal to the feature spacial dimensions and groups equal to the number of channels — group size equal to one. Since increasing group size allows inter-communication between channels; one might hypothesize that doing so could allow the squeeze step to extract richer representations of the input feature. To investigate such a hypothesis on OrthoNet, we conduct experiments using OrthoNet-34 with different grouping sizes. The results are recorded in Table 5.6. The results indicate that the group size is inconsequential; however, due to time and computational constraints, we were only able to run each experiment once, and more experiments need to be conducted.

Table 5.6: Effect of Grouping.

Method	Group Size	Top-1 acc
OrthoNet-34	1	75.13
	4	75.20
	$H \times W$	75.18

5.5.4 Fine-Tuning channel attention filters

In this section, we experiment with allowing the orthogonal filters to learn and fine-tune. We implement multiple different learning methods. First, we train OrthoNet-34 on constant orthogonal filters, then introduce learning the filters in the last twenty epochs. We call this method FineTuned-20. Second, we implement FineTuned-40, where we learn during each epoch that's divisible by five and also for the last twenty epochs — 40 epochs in total. Third, we implement learning the first 30 epochs — FineTuned-30 — then we disable learning the filters and continue training for the remaining 70 epochs on OrthoNet-50. Experiments demonstrate that learning the filters for OrthoNet-MOD-50 doesn't improve the overall validation accuracy. As for OrthoNet-34, we observe a more consistent accuracies for different learning methods. Results are shown in table 5.7. Further investigation is needed on the method of training and the potential inclusion of an attention-specific loss function for optimal learning.

Table 5.7: Effect of Squeeze Filter Learning.

Backbone	Learning Method	Top-1 acc
OrthoNet-34	FineTuned-20	75.20
OrthoNet-34	FineTuned-40	75.24
OrthoNet-50	FineTuned-40	78.50
OrthoNet-MOD-50	FineTuned-40	78.30
OrthoNet-MOD-50	FineTuned-30	78.36

5.5.5 Ease of Integration

Similar to SENet and FcaNet, our module can easily be integrated to any existing convolutional networks. The major distinction between SENet, FcaNet and our method is the adoption of different channel compression methods. As discussed earlier, our method can be described as a convolution as shown in 4 and can be implemented with a single line of code.

5.5.6 Limitations

Random filters have a natural limitation. In upper layers, where $HW > C$, it is impossible to have filters that comprise a full basis (i.e., you must choose C filters from the HW total). Although we did not observe it, a poor choice of filters may exist. Their unlearnable nature makes this a persistent factor throughout training. This could lead to a failure case. However; in the lower layers, we are able — and do — create a complete basis for $\mathbb{R}^{HW}$ which eliminates this issue in those layers.

Although we reported the means for OrthoNet-34 over five trials and OrthoNet-50 over four trials; due to limitations in time and computational capabilities, we only ran most other experiments once. Due to the limited number of runs, we could not report the standard deviation. However, OrthoNet's constant higher accuracy on a variety of tasks and datasets suggest the robustness of our network.

For the object detection and segmentation with MaskRCNN, we utilize the base configuration provided by MMDetection [41]. The results reported by the base configuration are different from those of their configuration. We were unable to use the configuration used by FcaNet [7] because of technical difficulties.

We also used the results of the experiments conducted by FcaNet for previous methods (FasterRCNN and ImageNet); however, we conducted our experiments under the same configuration.

5.6 Conclusion

In this work, we introduced a variant of SENet which utilizes orthogonal squeeze filters to create an effective channel attention module that can be integrated into any existing network. To evaluate its performance, we constructed OrthoNet and demonstrated state-of-the-art performance on Birds, Places365, and COCO with competitive or superior performance on ImageNet. By comparing OrthoNet to state-of-the-art, we've found that the key driving force to a successful attention mechanism is orthogonal attention filters. Orthogonal filters allow for maximum information extraction from any correlated features and allow the network to prescribe meaning to each channel yielding a more effective attention squeeze.

Our future works include further investigating learnable orthogonal filters, implementing metrics for evaluation of attention mechanisms, and further pruning our attention module to lower the computational cost and improve our method accuracy. We're also searching for a theoretical framework to explain the channel attention phenomenon.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015.

[2] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017.

[3] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 818–833, ISBN: 978-3-319-10590-1.

[4] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[5] H. Touvron, A. Vedaldi, M. Douze, and H. Jegou, "Fixing the train-test resolution discrepancy," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[7] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 783–792.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, 1998, pp. 2278–2324.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012.

[10] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[11] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015, pp. 730–734. DOI: 10.1109/ACPR.2015.7486599.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[13] S. Zagoruyko and N. Komodakis, *Wide residual networks*, 2016. DOI: 10.48550/ARXIV.1605.07146. [Online]. Available: https://arxiv.org/abs/1605.07146.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[15] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[16] X. Pan, C. Ge, R. Lu, *et al.*, "On the integration of self-attention and convolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 815–825.

[17] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[18] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[19] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[20] X. Song, Y. Dai, D. Zhou, *et al.*, "Channel attention based iterative residual learning for depth map super-resolution," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[21] Z. Zhong, Z. Q. Lin, R. Bidart, *et al.*, "Squeeze-and-attention networks for semantic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[22] X. Wang, L. Lian, and S. X. Yu, "Unsupervised visual attention and invariance for reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 6677–6687.

[23] V. VS, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel, "Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 4516–4526.

[24] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 4794–4803.

[25] R. K. Srivastava, K. Greff, and J. Schmidhuber, *Highway networks*, 2015. DOI: 10.48550/ARXIV.1505.00387.

[26] J. Fu, J. Liu, H. Tian, *et al.*, *Dual attention network for scene segmentation*, 2018. DOI: 10.48550/ARXIV.1809.02983.

[27] X. Wang, R. Girshick, A. Gupta, and K. He, *Non-local neural networks*, 2017. DOI: 10.48550/ARXIV.1711.07971.

[28] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, *Gcnet: Non-local networks meet squeeze-excitation networks and beyond*, 2019. DOI: 10.48550/ARXIV.1904.11492.

[29] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, *Rotate to attend: Convolutional triplet attention module*, 2020. DOI: 10.48550/ARXIV.2010.03045. [Online]. Available: https://arxiv.org/abs/2010.03045.

[30] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, *Cbam: Convolutional block attention module*, 2018. DOI: 10.48550/ARXIV.1807.06521.

[31] Z. Gao, J. Xie, Q. Wang, and P. Li, *Global second-order pooling convolutional networks*, 2018. DOI: 10.48550/ARXIV.1811.12006.

[32] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, *Eca-net: Efficient channel attention for deep convolutional neural networks*, 2019. DOI: 10.48550/ARXIV.1910.03151.

[33] H. Salman, C. Parks, S. Y. Hong, and J. Zhan, *Wavenets: Wavelet channel attention networks*, 2022. DOI: 10.48550/ARXIV.2211.02695. [Online]. Available: https://arxiv.org/abs/2211.02695.

[34] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[35] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, *Microsoft coco: Common objects in context*, 2014. DOI: 10.48550/ARXIV.1405.0312.

[36] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[37] G. Piosenka, "Birds 450 - species image classification," version 45, Feb. 2022. [Online]. Available: https://www.kaggle.com/datasets/gpiosenka/100-bird-species.

[38] G. Strang, "The discrete cosine transform," *SIAM Review*, vol. 41, no. 1, pp. 135–147, 1999. DOI: 10.1137/S0036144598336745.

[39] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[40] A. Paszke, S. Gross, F. Massa, *et al.*, *Pytorch: An imperative style, high-performance deep learning library*, 2019. DOI: 10.48550/ARXIV.1912.01703. [Online]. Available: https://arxiv.org/abs/1912.01703.

[41] K. Chen, J. Wang, J. Pang, *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark. arxiv 2019," *arXiv preprint arXiv:1906.07155*,

[42] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[43] Z. Zhuang, M. Tan, B. Zhuang, *et al.*, "Discrimination-aware channel pruning for deep neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[44] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.

[45] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

6 Chapter 6

Conclusion

In this doctoral dissertation, we have proposed and assessed a variety of machine learning and AI strategies for enhancing the categorization of single-view and multi-view photos. We have demonstrated that these strategies can increase the accuracy and robustness of classification algorithms in a variety of real-world circumstances.

First, we investigated an efficient and effective method for measuring the similarity of millions of unlabeled face photos. The suggested method is based on learning a discriminative embedding space using deep Siamese neural networks that capture facial features and their variations effectively. By conducting extensive trials on multiple benchmark datasets, we established the superiority of our method. Our approach offers a potential avenue for future research in this field, given the increasing relevance of artificial intelligence and computer vision.

Our second contribution was to enhance generative adversarial networks and present a novel method for grouping multi-view data that includes semi-supervised learning and feature fusion. The suggested method efficiently integrates labeled data and complementary information from multiple perspectives to improve clustering performance. The experimental findings on multiple real-world datasets reveal that the suggested method for multi view datasets is superior to the state-of-the-art methods. This study makes a significant contribution to the field of multi-view data clustering and can be used to other relevant fields.

Investigating channel attention mechanisms is another contribution of this dissertation. In our research, we provide a novel strategy for addressing the problem of channel attention by compressing spatial data via wavelet transform. By adding attention processes at several network layers, WaveNets are able to efficiently learn and represent complicated information through the use of various wavelet filters. The experimental results reveal that WaveNets can achieve state-of-the-art performance on the ImageNet dataset while maintaining an acceptable computational overhead. The WaveNets design has enormous promise for upgrading the state of the art in diverse

signal processing domains, such as image and speech processing, and could pave the way for new discoveries in these fields.

Our last contribution is an investigation into the influence of attention filter orthogonality in channel attention networks. By introducing orthogonal channel attention modules, we offer a unique architecture for boosting the performance of deep neural networks. The proposed modules permit the network to choose attend to informative channels, hence enhancing the model's precision and resiliency. Experiments performed on a variety of picture classification benchmarks illustrate the efficacy of the proposed method, which achieves high performance with fewer parameters and processing than previous methods. Future work include introducing metrics for channel attention, applying orthogonal channel attention with different design than squeeze and excitation.

Overall, our research reveals the capacity of machine learning and AI techniques to enhance the classification of single- and multiple-view photos. Some real-world applications, such as autonomous driving, surveillance, and robots, are applicable to the methodologies we have developed. We believe this study can inspire future research in this field and contribute to the creation of more accurate and robust categorization algorithms.