SOCIAL MEDIA IMAGE CLASSIFICATION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

ÇAĞRI UTKU AKPAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2017

Approval of the thesis:

SOCIAL MEDIA IMAGE CLASSIFICATION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

submitted by **ÇAĞRI UTKU AKPAK** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____

Prof. Dr. Ferda Nur Alpaslan
Supervisor, **Computer Engineering Department, METU** _____

Examining Committee Members:

Prof. Dr. Halit Oğuztüzün
Computer Engineering Department, METU _____

Prof. Dr. Ferda Nur Alpaslan
Computer Engineering Department, METU _____

Prof. Dr. Ahmet Coşar
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU _____

Assist. Prof. Dr. Orkunt Sabuncu
Computer Engineering Department, TED University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ÇAĞRI UTKU AKPAK

Signature :

ABSTRACT

SOCIAL MEDIA IMAGE CLASSIFICATION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Akpak, Çağrı Utku
M.S., Department of Computer Engineering
Supervisor : Prof. Dr. Ferda Nur Alpaslan

September 2017, 39 pages

Increasing popularity of social media platforms has led to an increase in the number of unclassified images. Given the complexity of images uploaded to these platforms and the number of classes available, it is clear that traditional image classification methods are not suitable for this kind of classification. Previous research on this topic primarily focuses on Deep Neural Networks to overcome the limitations of traditional methods. In these studies, researchers either limited the scope of their dataset; for example, handwritten digits, or combine their approach with Natural Language Processing methods to create meaningful descriptions. Similarly in this study, we use Deep Convolutional Neural Networks to classify social media images. Unlike previous approaches, there is no limitation on the scope of the images and classes represent textual tags that explain images in a simple and natural way. Moreover, the previous approaches do not allow class expansion after the training. To overcome this difficulty, a modular system is developed for classification. Separate networks are trained for each

individual class and they are combined to create the overall system. Using this system new classes can be introduced without affecting the performance of the previously trained classes. Experiments are done on a dataset complied from social media platforms and this approach achieves promising results.

Keywords: Image Classification, Artificial Neural Network, Convolutional Neural Network, Deep Learning

ÖZ

DERİN KONVOLÜSYONEL SİNİR AĞLARIYLA SOSYAL MEDYA RESİMLERİ SINIFLANDIRMASI

Akpak, Çağrı Utku
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Ferda Nur Alpaslan

Eylül 2017 , 39 sayfa

Sosyal medya platfromlarının yaygınlaşmasıyla beraber sınıflandırılmamış resimlerde bir artış görünmektedir. Bu platformlara yüklenen resimlerin çok detaylı olması ve yapılabilecek sınıflandırmaların çok sayıda olmasından dolayı, bu tür bir sınıflandırmayı geleneksel resim sınıflandırma yöntemleriyle yapmak açık bir şekilde mümkün görünmemektedir. Bu konu üzerindeki önceki araştırmalarda bu problemi çözmek için derin sinir ağları kullanılmıştır. Araştırmacılar ya kullanılan resimlerin sınıflandırma kapsamını daraltmışlardır ya da doğal dil işleme yöntemleriyle birlikte derin sinir ağlarını beraber kullanarak resimlere anlamlı açıklama üretmişlerdir. Benzer şekilde bu araştırmada, biz derin konvolüsyonel sinir ağlarını kullanarak sosyal medya resimleri üzerinde sınıflandırma işlemi gerçekleştirdik. Önceki araştırmaların aksine, sınıflandırma kapsamında herhangi bir kısıtlandırma yapılmamıştır ve resimleri etiketlemek için kullandığımız sınıflar, resimleri yalın ve doğal bir şekilde anlatmaktadır. Buna ek olarak, önceki

araştırmalar baştan tanımlanmış sınıfları arttırmaya izin vermemektedir. Bu zorluğun üstesinden gelmek için, modüler bir sınıflandırma sistemi geliştirilmiştir. Her bir sınıf için ayrı bir sinir ağı eğitilmiş ve bu ağlar birleştirilerek tüm sistem oluşturulmuştur. Bu sistemi kullanarak, sisteme önceden tanımlanmış sınıfların performansını bozmadan yeni sınıf eklemek mümkün hale gelmiştir. Sosyal medya resimlerinden toplanmış veri kümesi üzerinde yapılan deneylerde bu yaklaşım, gelecek vadeden sonuçlar elde etmiştir.

Anahtar Kelimeler: Resim Sınıflandırma, Yapay Sinir Ağları, Konvolüsyonel Sinir Ağları, Derin Öğrenme

To my mother

ACKNOWLEDGMENTS

First of all, I would like thank my supervisor, Prof. Dr. Ferda Nur Alpaslan, for advising me throughout my study and taking time during her vacation to help me complete my thesis. I am also grateful for the machine learning course I have taken from her that introduced me to the field.

I would also like to thank Özgür Alan whom provided me with the topic and motivated me to work on this project. Our discussions provided me with insight and allowed me to overcome the problems I have encountered. This thesis would not be possible without him.

During the difficult parts, my friends eased my fears and encouraged me to finish my thesis. They were always there for me and I cannot thank them enough.

Finally, I would like to thank my mother. Her continuous support and encouragement throughout my entire life has been invaluable. While writing my thesis, she made it possible for me to forget about everything and just focus on my thesis. I am extremely lucky to have her in my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
ML	Machine Learning

CHAPTER 1

INTRODUCTION

Image classification is a commonly tackled problem in Artificial Intelligence (AI). Although great variety of classification problem exists; we focus on the classification of images submitted by users to social media platforms. The classes represent the textual tags to label the images with. They must be in a human-readable format and explain the image in a natural and practical way, for example, birthday party instead of four humans eating a cake. With the growing use of social media platforms and accessibility of mobile phones with camera capabilities, number of unclassified images are rapidly rising. Given the detail and complexity of the images uploaded to social media platforms, it is becoming more apparent that traditional approaches to images classification is not enough to solve this problem. The traditional image classification methods generally focus on hand crafted features to do this task. First, hand-crafted features are extracted from the image and then AI methods are applied to the extracted features. The features are generally extracted using computer vision [19]. Although this approach garnered successful results on simple images with limited features, it is not enough to correctly classify complex images usually found on social media platforms. When the complexity of the image and the number of classes increase, it is impractical to rely on hand crafted features of the image for classification. Moreover, with the increased complexity and detail of the images, the amount of features required for correct classification increases significantly and requires too much effort for extraction.

Deep learning is the solution to these problems. It is an advanced Machine

Learning (ML) technique that uses deep Artificial Neural Networks (ANN). Although various forms of ANNs have been used since 1960s, deep networks have recently become more common. There are two reasons for the recent popularity of Deep Learning. First one is the growing computational power of processors and new parallel computation focused units (e.g. GPUs, Neural Processors, Cloud Server Farms, etc). ANNs require significant amount of extra computation for training with each increasing depth. This means that in the early days, training took too long for deep ANNs to be practical. Second one is the amount of data available to use for training. Deep ANNs need large amount of data to accurately learn the required application and to prevent over-fitting. Social media platforms and growing use of cheap technological devices with camera capabilities solved the data problem that is present in the early days of ANNs.

There are several advantages of using Deep Learning. First one is the elimination of the feature extraction step in the classification process. Instead of hand-picked features, deep ANNs can extract features directly from raw data (or images in this case) for classification. Then classification can be done directly on the deep ANN or extracted features can be used with other AI methods. Elimination of feature extraction relieves the effort of finding good features and makes the system easier and more practical. Moreover, finding features and extracting them can be nearly impossible on a complicated image. Second advantage comes from the layered structure of ANNs. Each layer provides additional level of abstraction and increases the complexity of derived features. This means that a deep ANN can extract very high level features automatically from the image and use it for classification.

There are two difficulties that we have identified and need to address in this research. First and the most important one is the ability to correctly classify the images. Unlike other classification problems that are focused on (very specific set of images like handwritten digits (MINST)), this problem is more broad and general purpose. The broadness of the images and vast differences between images in the same class makes it very difficult to correctly classify images. Second difficulty comes from the sheer number of classes that exists in user tagged images on social media platforms, introducing and correctly classifying

new classes and, difficulty of correctly classifying images before and after the introduction of new classes. Previous researchers on this topic either limited the scope of the images to avoid this problem or focused on the objects within the image together with their actions and properties and using Natural Language Processing (NLP) methods to create descriptions. Both of the solutions are not suitable for our approach, because we want practical human-readable tags that can be used on majority -if not all- of the images. Limiting the scope of the images will not be useful and practical in the ever changing nature of social media and created descriptions using Deep Learning and NLP are not very human-readable even though they are very accurate.

We have found two solutions to the problems defined above. For the first problem, we use a 6 layer Convolutional Neural Network (CNN) for direct classification of images with their respective classes. The reason we selected CNN for the architecture is stemmed from the fact that previous approaches to image classification and recognition achieved their best results using this architecture and it is very advantageous for visual applications. After all the experiments we have conducted, we achieved best results on three different networks with different parameters. Based on these results, we have achieved 21.57% validation and 13.73% test error rates with images scaled to 64×64 pixel dimensions on two classes on our best network. However these results are limited to two classes we have previously defined in the dataset and does not allow for expansion. Despite this limitations, this preliminary results shows that complex images can be classified directly using this approach. For the second problem, we use a modular approach for classification. For each class a separate neural network is trained and these networks are combined to create a system that can classify multiple classes. This approach allow for new class introduction without affecting the performance of other classes and speeds up the training speed due to the simplicity of each network. Since every network only classifies one class, simpler network architectures can be used and trained in a relatively short time. Using similar 6 layer architectures from the first solution, we have achieved achieved classification accuracy of 66.31% on the overall system and 71.74% on the best single class network.

Outline of the thesis is as follows. Literature review methods for image classification is given in the chapter 2. In chapter 3, background information about ANN, CNN and deep learning is provided. Experimental information about the data-set, the method and the results are given in chapter 4.

CHAPTER 2

LITERATURE REVIEW

Previous research on image classification can be divided into three categories. The first approach focuses on low level features like colour, shape or texture[21, 16, 25, 10, 7, 11, 12, 27]. However this low level features does not provide enough information to correctly classify images. Instead, they focus on image annotation which is assigning certain meta-data information to the images. Moreover, this approach examines the details of the images instead of focusing on the overall picture as we did in our research.

Second approach attempt to learn the high level concepts within an image for correct classification. These approaches focuses on high level detail together with probabilistic models within an image to correctly label and classify them. They require large number of training samples and uses concept models to correctly annotate images. These methods include using multiple bernoulli reference models for word annotation [8], Support Vector Machines for large image classification and labeling[6], and constraining the latent space with Probabilistic Latent Semantic Analysis models[18] to achieve results. Although this work offer promising results, the main focus of this research is Information Retrieval purposes which aims fast indexing, searching using indexes. Moreover, it generally does not offer human-readable and practical textual tags that can be used as classes required to be used in social network applications.

Final approach is the image classification using neural networks. This approach can be divided into three categories. First category is about general purpose networks. Previous research of this category include parameter optimiza-

tion of network[13] to optimize previously constructed architectures to increase their performance and convolutional neural networks on Imagenet dataset[15] to achieve best results. Although they have achieved successful classification or at least improve the performance of state-of-the-art results, the architecture used in these papers are very complicated which in turn limits their practicality. Moreover these networks have fixed number of classes that they can classify and does not allow class expansion. Neural networks are also used for content based classification by dividing the objects in the image between foreground and background[20]. In this paper, shape based features are extracted from wavelet-transformed images. These features are extracted using a neural network and classified directly. Using a dataset of 300 training data and 300 test images divided equally between 30 classes shows classification rates of %81.7 and %76.7. Although these papers offer great results with error rates as low as %18.7 percent on a known dataset like Imagenet, the classes of this dataset and classification results are very simple and they are not suitable to be used in social media applications where the images are fairly complicated.

Research on the second category aims to generate meaningful descriptions of the images using a combination of neural networks and NLP techniques. Although this is not a classification, the idea behind is similar to our approach which is to label images in a natural and understandable way. This is a fairly new approach, we can only find two example research in this category. First one[14] uses a combination of encoder and decoder pair to create novel descriptions for the images. Encoder learns the image-text representations from the data and decoder uses a novel language model to decode these representations to natural language. These descriptions are generated from scratch and the encoded representations are only used to rank images and their sentences. The encoder used in this study, is a Long Short-Term Memory which is a type of recurrent neural network and using this encoder state-of-the-art performance has been reached on Flickr8K and Flickr30K datasets. In the second paper[26], the researchers uses generative model that is based on recurrent neural network architecture to classify images to their descriptions. The training is done to maximize the likelihood of the description and achieved accurate results which measured qualitatively

and quantitatively. In the pascal dataset, this paper achieves the BLEU score (score to determine the quality of sentences) of 59 which surpasses state-of-the-art score of 25 and comes very close to human performance of 69. Moreover it also exceeds state-of-the-art accuracy on Flickr30k and SBU datasets. Although these approaches achieves successful results, the generated descriptions are fairly descriptive and complicated. Instead, we aim to make simple and natural labels using the classification results of our classes.

Limiting the scope of the images for classification is the third category in neural network based classification. In this category, scope of the images are fixed to a certain field and results are aimed at a clear objective based around these fields. This approach is the most commonly used method in real-life applications in automation and biological fields. In the first paper[4], the researchers achieved a superhuman performance on German traffic sign recognition benchmark of 99.46% accuracy. They described a model that directly classifies traffic signs without extracting any features and boost its performance using different pretrained multimodal networks on different parts of the dataset. Similar to the previous paper, multimodel convolutional neural networks are also used in the MINST and traffic sign recognition datasets in the paper by Dan et al.(2012)[2]. This study becomes the first study to achieve near-human performance on MINST dataset and outperforms human performance on traffic sign benchmark. In the biology field, neural networks have been used to segment neural membranes in electron microscopy images[3] and to detect breast cancer mitosis on breast cancer histology images[5]. In the first study, deep convolutional network is used to predict the segmentation of pixel values from the raw pixel values directly (without any post-or preprocessing) to classify neuron membranes in the brain. Without using any post processing methods, this study won the International Symposium on Biomedical Imaging (ISBI) 2012 EM Segmentation Challenge by a large margin and even exceeded human performance in some aspects of the field. Similarly in the second study, researchers used deep convolutional neural network to classify each pixel of the histology images with small post-processing at the end. Classification aims to find the center of the cell that are undergoing mitosis. Researchers of this study won the International

Conference on Pattern Recognition (ICPR) 2012 mitosis detection competition with a large margin from other competitors.

CHAPTER 3

BACKGROUND

In this study, we use a deep CNN which is a type of Artificial Neural Network (ANN) for social media image classification. Like all ML methods it improves its performance or "learns" a task over successive iterations from a dataset. For example, learning the location of faces in an image using example images together with additional information about the location of faces in it. This differs from rule-based methods where the performance of the system depends on the programmed rules for a very specific task. Machine Learning methods can be used to learn different tasks based on its examples which makes them more flexible than rule-based system. Information about the details of these method are given in the following sections. Recent research on these topics can be found in chapter 2.

3.1 Artificial Neural Network

ANNs are special form of networks inspired from the nervous system of animals. They consist of artificial neurons which model the axioms that are present in the nervous systems. They are utilized to approximate complex nonlinear mathematical functions that involve large number of variables (or inputs). The data model takes the form of weighted directed graph with activation functions, where each node takes the role of "neuron" and connects to other neurons. These neurons pass information between one another using their edges. In general, these neurons are ordered into layers for simplicity and control over their role in the system. Each layer represent a mathematical transformation on the output of

the previous layer. First layer represent the input to the entire system and the last layer's output represent the system's final output. Moreover Neurons can have a state value which is generally represented with real numbers and use this value for their calculations.

3.1.1 Perceptron and Backpropagation

Perceptron architecture is the first and simplest form of a neural network with only two layers. The input layer connects directly to the output layer and they can have multiple neurons.

Perceptron network uses perceptron learning rule to adjust its weights. This weight adjustment is done by calculating the difference between the desired and the actual network output. It utilizes this difference together with a learning rate to update the weights. The learning rule can be written using these equations:

$$y_j(t) = [w\vec{(t)} \cdot x_j], \tag{3.1}$$

$$w_i(t+1) = w_i(t) + \alpha(d_j - y_j(t))x_{j,i}, \tag{3.2}$$

where α is the learning rate, y_j is the output of the j^{th} neuron, w_i is the weight of the i^{th} neuron and d_j is the desired output.

It has been shown that perceptrons[17] could only solve linearly separable problems. In backpropagation model, additional layers (hidden layers) can be added and the discrete thresholding function can be changed for a continuous (sigmoid) one for complex non-linear functions. But the most important functionality of backpropagation is the generalized delta rule, which allows for adjustment of weights leading to the hidden layer neurons in addition to the usual adjustments to the weights leading to the output layer neurons. Using the generalize delta rule one can adjust the weights leading to the hidden units by backpropagating the error-adjustment.
Backpropagation Learning Rule:

$$\frac{\partial E}{\partial w_{ij}} = \delta_j o_i. \tag{3.3}$$

Then,

$$\delta_j = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial \mathrm{net_j}} = \begin{cases} (o_j - t_j)\varphi(\mathrm{net}_j)(1-\varphi(\mathrm{net}_j)) \\ \text{if } j \text{ is an output neuron,} \\ (\sum_{l\in L}\delta_l w_{jl})\varphi(\mathrm{net}_j)(1-\varphi(\mathrm{net}_j)) \\ \text{if } j \text{ is an inner neuron.} \end{cases} \tag{3.4}$$

Learning rate which is represented with α, is used to calculate the weight difference in gradient descent. It controls how much effect each iteration of the algorithm has, to the weights. The change in weight is calculated using this formula:

$$\Delta w_{ij} = -\alpha\frac{\partial E}{\partial w_{ij}}, \tag{3.5}$$

where E is the squared error, o is the output, φ is the calculated output. To update each weight, this simple formula is used:

$$w_{t-1,ij} = w_{t-1,ij} + \Delta w_{ij}, \tag{3.6}$$

where w_{t-1} and w_t represents old and new weights respectively.

3.2 Deep Neural Networks and Types

A deep neural network (DNN) [1, 23, 9] is an artificial neural network (ANN) with multiple hidden layers of units between the input and output layers. Similar to shallow ANNs, DNNs can also model complex non-linear functions. The extra layers enable feature composition from lower layers and has more potential for complex data modeling without the need for a feature extraction over shallow networks[1].

Feedforward Neural Network: A feedforward neural network is an artificial neural network where nodes are connected from layer to layer and there are no backward loops.

The feedforward neural network was the first discovered model and it has a simple composition. In feedforward networks, calculations are done in only one direction, which is forward, from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network. Error is backpropagated from the output nodes to the hidden nodes.

Convolutional Neural Network: CNN takes its name from the convolutional layers (see Convolutional Layer). It is very similar to Feedforward Neural Networks and each successive layer is fully connected to the next one. Convolutional layers consists of multidimensional filters with shared weights. This property use 2D or 3D structure of input data to its advantage; therefore, CNN surpasses other architectures in certain applications like image classification and speech recognition. Moreover, this architecture can accomplish tasks that are not possible with other architectures, namely DeepDream[24]. CNN's structure allow backpropagation to be used for training and have fewer parameters to estimate than other DNNs, making them very effective and popular [22, 15]. We have also used Deep CNN is this research and more details of this architecture are given in chapter 4.

The method we are using in this study is a deep CNN. It consists of convolutional, pooling and regular feedforward layers and log likelihood with logistic regression at the final layer for final classification. Brief information of convolutional and pooling layers are provided in the following subsections.

3.2.1 Convolutional Layer

A Convolutional layer consists of a set of filters. Each filter span a small area but cover the whole depth field of the input. For a forward pass each filter positioned horizontally and vertically on each input and dot product of the of the weight of the filter and area at the current position of the input are calculated. This allows for spatial features extracted from every part of the spatial space. Three hyperparameters determines the size of the output dimension in a convolutional layer: the depth, stride and zero-padding.

- Depth of the output determines the number of neurons in the layer that connect to the same region of the input. These neurons theoretically will activate for different features of the input and pass their calculations to the next layer.
- Stride determines how each filter positioned between one another. Smaller stride values results in overlapping fields and increases the output volume.
- Zero padding pads the borders of the input with zeros for precise control of the input volume.

3.2.2 Pooling Layer

Pooling Layer provides the functionality of nonlinear down-sampling. Max pooling is the most common form where input image is divided into sub-regions and maximum in each region is selected as the output. Although exact location of the features are lost during this procedure, relative locations are preserved which is crucial. It also reduces the size of the representation, in turn reduces the calculations and prevent overfitting of the input. It is commonly inserted after the convolutional layer which we also did in our network architecture.

CHAPTER 4

METHOD AND RESULTS

In this chapter, experimental results are explained in two parts.

4.1 Image Classification with Fixed Classes

For the first part of the study, we have used a dataset consisting of fixed number of classes. These results shows that complex social media images containing multiple details can be classified with a comparatively good accuracy. Details of the results are given in the following subsections.

4.1.1 Dataset

We used a dataset of images consists of two classes namely Birthday and Graduation which consist of 215 and 203 images respectively. For uniformity, all images are scaled to a three dimensional vector of $128 \times 128 \times 3$ and $64 \times 64 \times 3$ pixels where each pixel value is represented in the normalized RGB spectrum. Discrete RGB valeus divided to 256 to convert them into real between 0.0 and 1.0 for normalization. Output values are represented with two binary values 0.0 and 1.0 where they represent the probability of the image being in the respective class. Assuming n number of images in the dataset, they are divided into $\frac{n}{2} \mid \frac{n}{4} \mid \frac{n}{4}$ sized parts for training, validation and test phases. Sample images from Birthday and Graduation classes can be seen in Fig.4.1 and Fig.4.2 respectively.

Figure 4.1: Sample images for Birthday

4.1.2 Method Details

In order to classify images, we have used 6-layer convolutional neural networks. Experiments consists of training, validation and test phases. Training is done using backpropagation algorithm and gradient descent. Network architecture and hyper-parameters of the network are selected based on the previous studies and the experimental results. After network architecture and its hyper-parameters are established (see subsection 4.1.4), raw pixel values of the re-sized images (see subsection4.1.1) are fed directly into the network. The output is a 2-dimensional vector where each value is the probability of the object being in one of the classes. Negative log-likelihood function is then used to calculate the loss using the output vector from the network and the actual probability for that input. Training is stopped after certain epochs. At fixed intervals during training, which is de-

Figure 4.2: Sample images for Graduation

termined by validation frequency hyper-parameter, the validation percentage is calculated. If the validation error improved by a certain margin which is represented with the improvement threshold hyper-parameter, the network is selected for testing. Validation and test errors are calculated by dividing the number of wrongly classified images to the total number of images for both categories. Classification of the image is done by simply selecting highest class probability.

4.1.3 Evaluation Metric

We have two evaluation metrics in this study. First one is the loss value, which is calculated using negative log-likelihood at each epoch. This value is indicative of how well the network converges during training. Second one is the validation and test classification error values at each validation phase. These values are

calculated by dividing the number of wrongly classified images to the total for that category. The test value is only calculated if the validation error is lower than the previous best as it is described in section 4.1.2.

4.1.4 Networks Architectures and Results

We have achieved comparatively good results on three different networks. Majority of hyper-parameters of the networks are kept same, only filter counts of their convolutional layers and input sizes changed between networks. Filter sizes in the first layer are also changed to accommodate the new input dimensions. Their parameters and the experimental results are given in the following tables and graphs for each network. Different values between networks are highlighted in their respective tables.

Table 4.1: Network1 architecture and parameters.

Learning Rate	0.01
Number of Epochs	2500
Improvement Threshold	0.995
Validation Frequency	250
Pool Function	Max Pooling
Input Dimensions	**Width: 64** **Height: 64** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $33 \times 33 \times 3$ **Filter Count: 10** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $9 \times 9 \times 10$ **Filter Count: 25** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

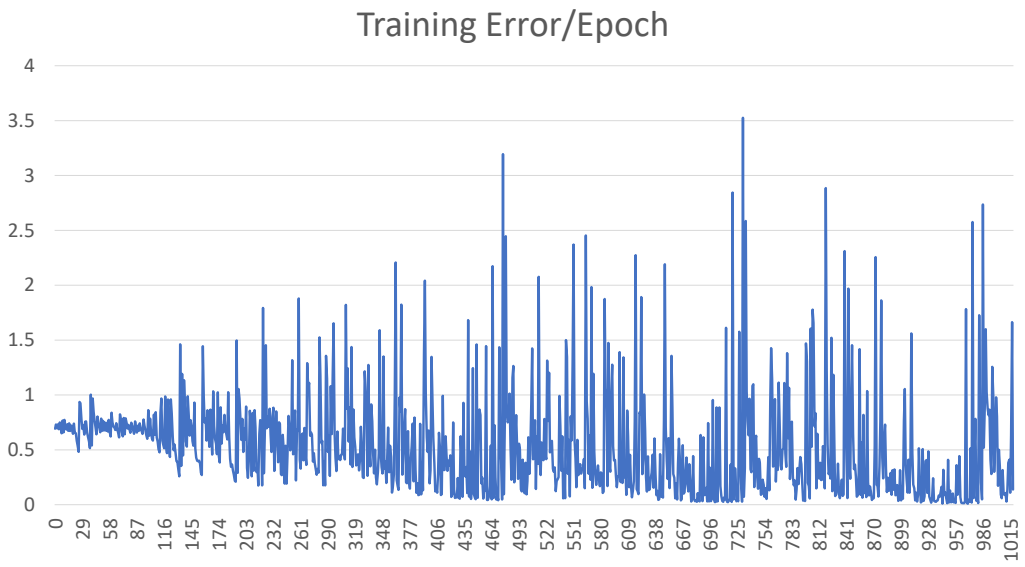

Figure 4.3: Training Error/Epoch for Network1

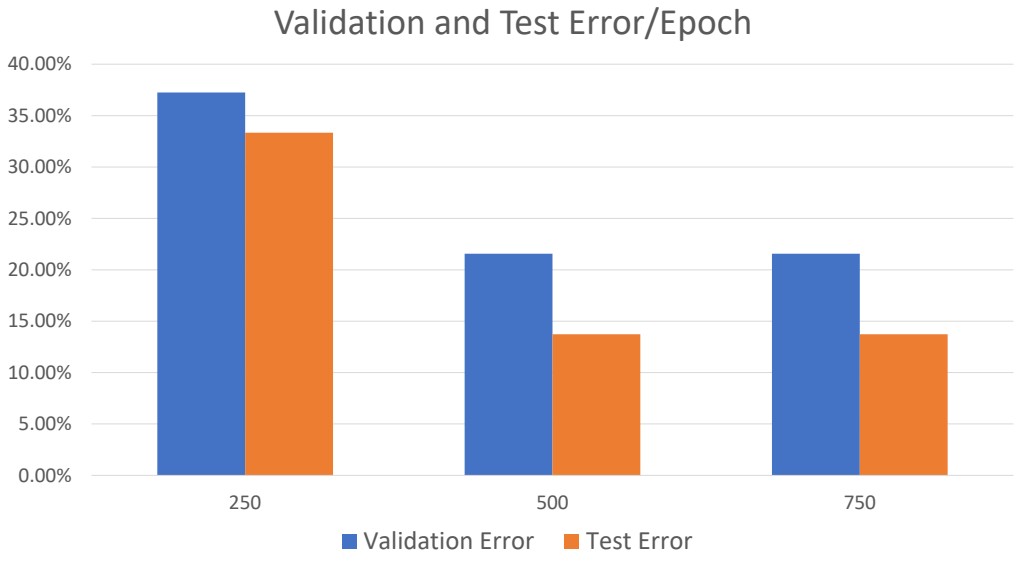

Figure 4.4: Validation and Test Error/Epoch for Network1

In Fig.4.3, we can see a fluctuating loss between values 1.0 and 0.5 with some

occasional spikes. The cause of these spikes determined to be the small size of the dataset. Each sample signifies an important training example and therefore has the ability to cause fluctuations. This further proved from Fig.4.4 where validation and test errors are continually decreasing after each validation. Considering the size of the dataset, this stability is taken as a unstable convergence. Best results are achieved using the smallest network (can be seen in Table 4.1). Because of the small size of the dataset, stable convergence cannot be established without sacrificing validation and test accuracy. This is later shown with experimental results below.

Table 4.2: Network2 architecture and parameters.

Learning Rate	0.01
Number of Epochs	2500
Improvement Threshold	0.995
Validation Frequency	250
Pool Function	Max Pooling
Input Dimensions	**Width: 128** **Height: 128** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $65 \times 65 \times 3$ **Filter Count: 10** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $17 \times 17 \times 10$ **Filter Count: 25** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

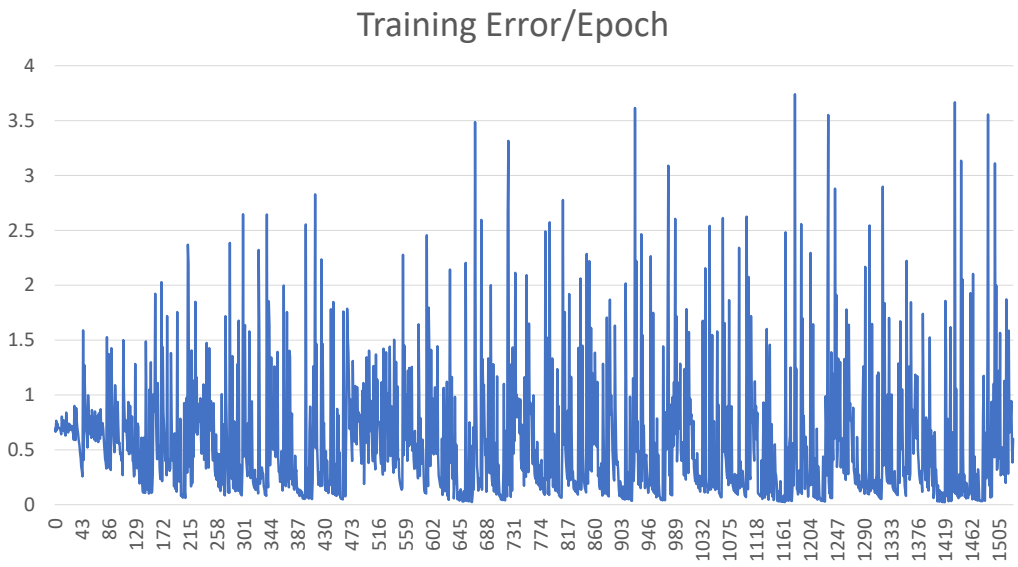

Figure 4.5: Training Error/Epoch for Network2

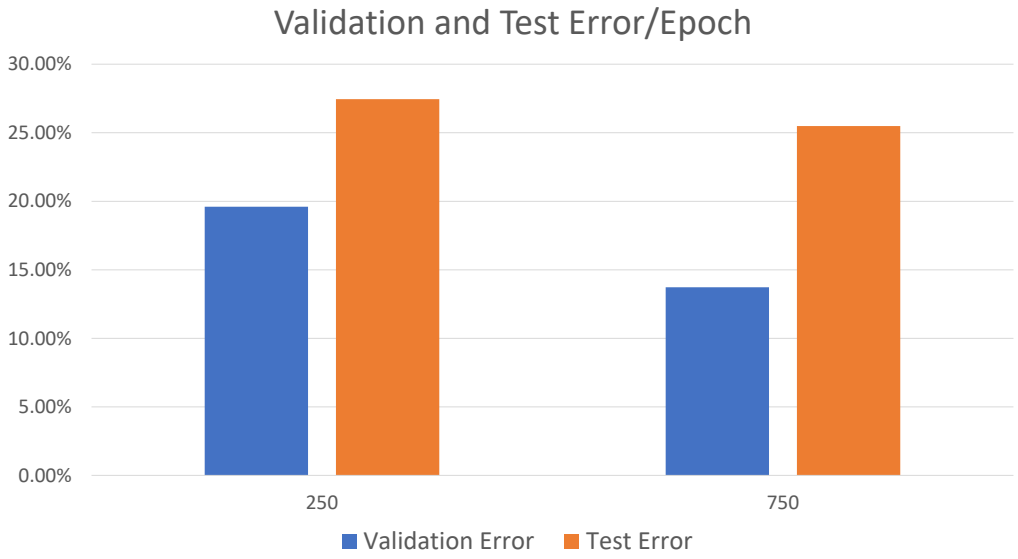

Figure 4.6: Validation and Test Error/Epoch for Network2

In this network (see Table 4.2), input is twice the size of the previous network.

However filter counts in the first and second convolutional layers are kept same. The reason for this is to observe the changes in the validation and test errors with a bigger input. The results are not satisfactory. In Fig.4.5, we again see a fluctuating loss values. However considering the previous results, the fluctuations are more pronounced. Moreover, we can see that there is an increase in validation and test errors from Fig.4.6. This led us to believe that increase the size of the input without compensating with increased filter counts results in a lower accuracy in validation and test results.

Table 4.3: Network3 architecture and parameters.

Learning Rate	0.01
Number of Epochs	2500
Improvement Threshold	0.995
Validation Frequency	250
Pool Function	Max Pooling
Input Dimensions	**Width: 128** **Height: 128** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $65 \times 65 \times 3$ **Filter Count: 50** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $32 \times 32 \times 50$ **Filter Count: 125** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

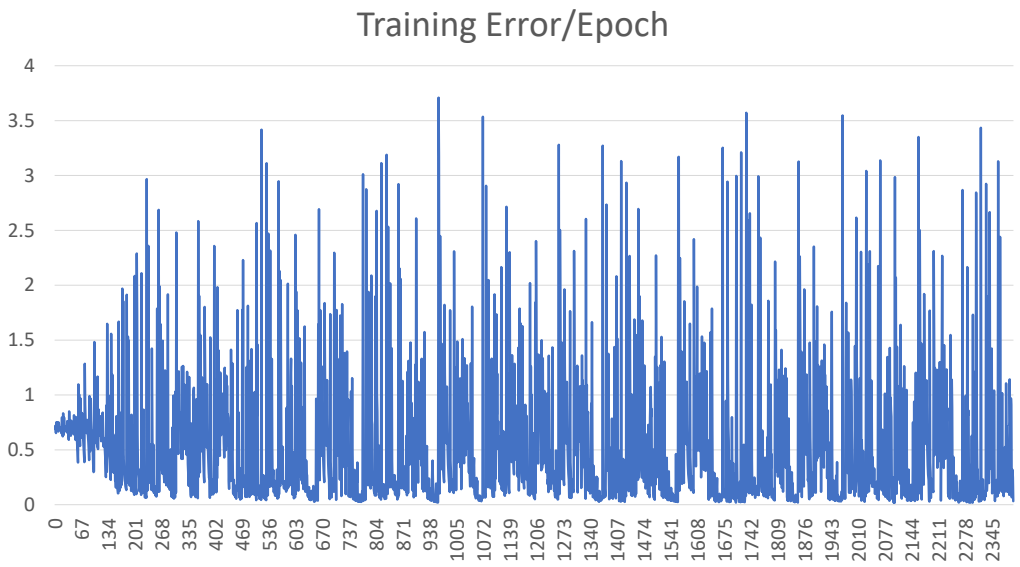

Figure 4.7: Training Error/Epoch for Network3

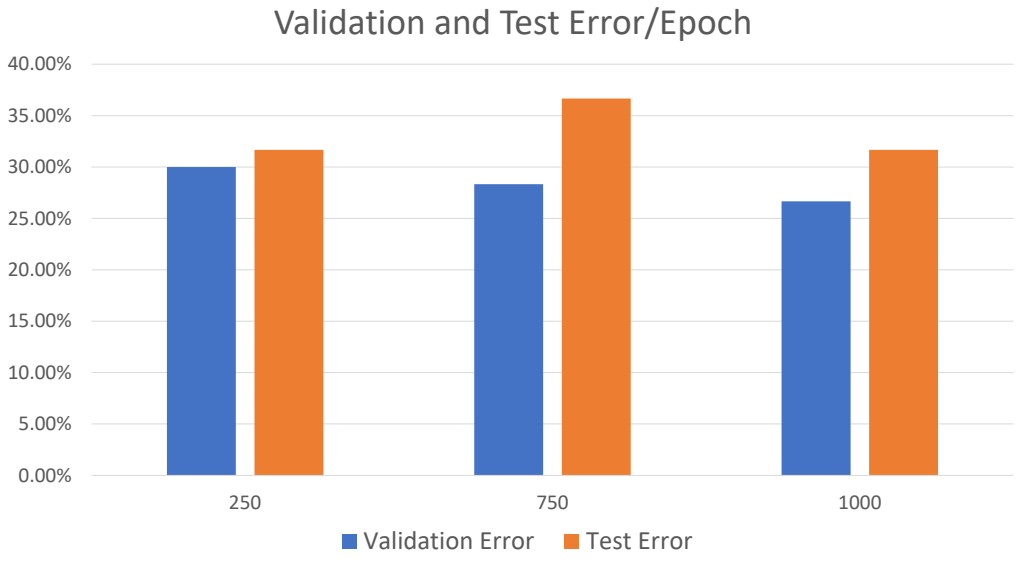

Figure 4.8: Validation and Test Error/Epoch for Network3

In response to the previous experiment, the filter counts in the convolutional

layer are also increased to compensate for the increase in the input sizes in this network (can be seen in Table 4.3). Although in Fig.4.7, we can see fluctuating loss values are also pronounced like the results from network2 (see Fig.4.5, validation and test errors are decreased (as seen in Fig.4.8). This led us to believe that with a larger dataset, increasing the input size and filter counts can achieve better and more stable results.

4.1.5 Evaluation

In Figures 4.3, 4.5, 4.7 we can see that loss value does not have stable convergence and fluctuates around 0.5. However, we observe that validation and test errors are decreased or remained same at each validation. Fig. 4.4 shows that we have achieved our best result with 21.57% validation and 13.73% test errors at 750 epoch using the network1 architecture. Although training continued after 750 epoch, the results did not change.

4.2 Modular Image Classification with Expandable Classes

For the second part of the study, we have used a modular system that can classify each class in the dataset semi-independently. For each class separate network is trained and these networks are combined to create the overall system. Details of the results are given in the following subsections.

4.2.1 Dataset

We used a dataset of images consists of five classes in this part. Dataset composition can be in Table 4.4 and sample images for the remaining three classes can be seen in Figures 4.9, 4.10 and 4.11. Normalization and scaling of images is done in the same way as the previous section (see subsection 4.1.1). However since each class needs to be trained separately, we have created five different subsets to train each class with values for positive and negative targets. To keep each subset balanced, equal number of positive and negative training, test and validation samples are combined. For each class subset, positive samples are taken from the class images and negative samples are taken in equal amounts

from other class images. For example, for birthday training sample subset, 200 images in the positive samples are matched with 50 negative samples from remaining four classes. Images in the dataset kept independent from another to avoid class cohesion.

Table 4.4: Dataset composition.

Class	Size
Graduation	200
Selfie	192
Festival	116
Picnic	160
Birthday	216

Figure 4.9: Sample images for Festival

Figure 4.10: Sample images for Picnic

4.2.2 Method Details

Training, validation and method details are same from the previous section (for details see subsection 4.1.2). Training, validation and test of each class is performed independently. For the classification on the overall system, each image is given as an input to every class network. Final class is selected from the representative network with highest positive probability.

4.2.3 Evaluation Metric

There is only one metric used in the part of the experimental results. It is the independent final validation and test error rates for each class. Weighted validation and test error rates for the overall system are also calculated to determine

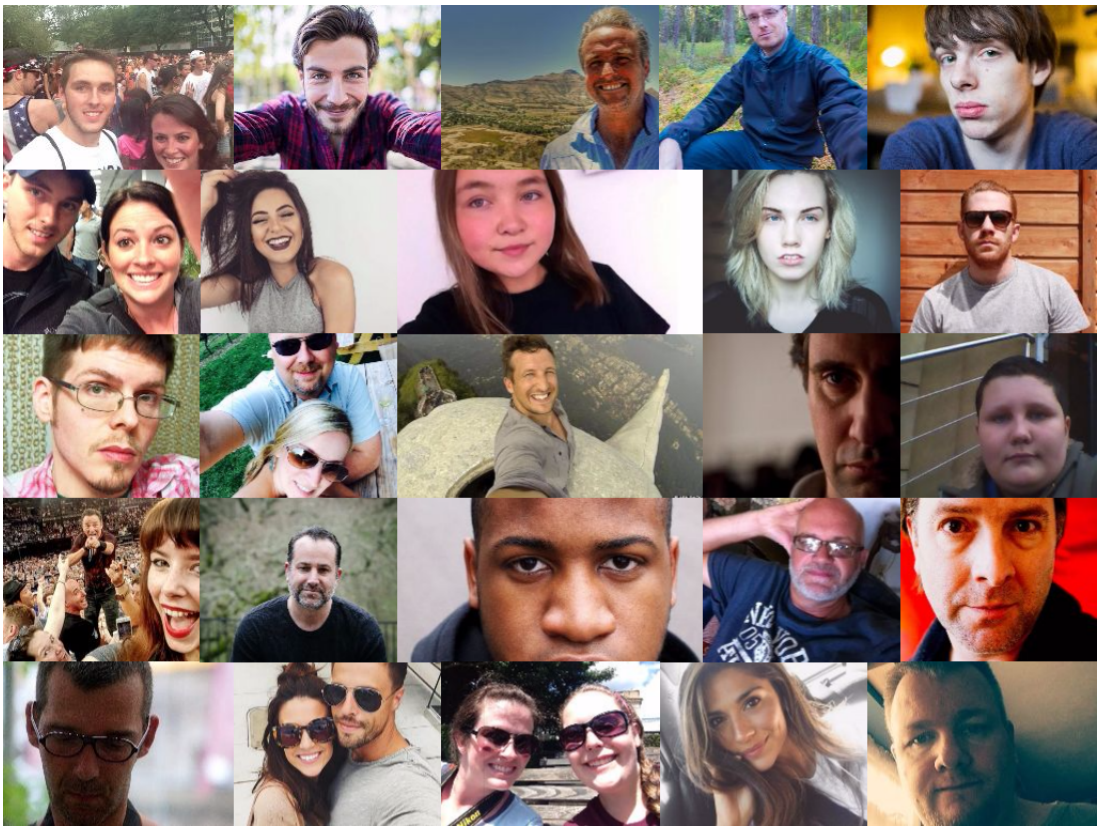

Figure 4.11: Sample images for Selfie

the overall accuracy.

4.2.4 Class Network Architectures and Classification Accuracy

We have used same architecture for each class network and optimize the hyper-parameters to achieve lowest validation and test error in each network. Their parameters and the experimental results are given in the following tables and graphs for each network. Different values between networks are highlighted in their respective tables.

Table 4.5: Graduation network architecture and parameters.

Learning Rate	**0.01**
Pool Function	Max Pooling
Input Dimensions	**Width: 64** **Height: 64** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $33 \times 33 \times 3$ **Filter Count: 50** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $16 \times 16 \times 50$ **Filter Count: 125** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

Table 4.6: Selfie network architecture and parameters.

Learning Rate	**0.01**
Pool Function	Max Pooling
Input Dimensions	**Width: 64** **Height: 64** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $33 \times 33 \times 3$ **Filter Count: 50** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $16 \times 16 \times 50$ **Filter Count: 125** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

Table 4.7: Festival network architecture and parameters.

Learning Rate	**0.01**
Pool Function	Max Pooling
Input Dimensions	**Width: 64** **Height: 64** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $33 \times 33 \times 3$ **Filter Count: 10** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $16 \times 16 \times 10$ **Filter Count: 25** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

Table 4.8: Picnic network architecture and parameters.

Learning Rate	**0.03**
Pool Function	Max Pooling
Input Dimensions	**Width: 64** **Height: 64** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $33 \times 33 \times 3$ **Filter Count: 10** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $16 \times 16 \times 10$ **Filter Count: 25** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

Table 4.9: Birthday network architecture and parameters.

Learning Rate	**0.02**
Pool Function	Max Pooling
Input Dimensions	**Width: 64** **Height: 64** **Depth: 3**
1. Layer	Type: Convolutional **Filter Shape:** $33 \times 33 \times 3$ **Filter Count: 20** Activation Function: Tanh
2. Layer	Type: Pooling Pool Size: 2×2
3. Layer	Type: Convolutional **Filter Shape:** $16 \times 16 \times 20$ **Filter Count: 50** Activation Function: Tanh
4. Layer	Type: Pooling Pool Size: 2×2
5. Layer	Type: FeedForward Neuron Count: 50 Activation Function: Tanh
6. Layer	Type: FeedForward Neuron Count: 2 Activation Function: Tanh

From these tables, we can observe two points. First, best results are achieve with input sizes of 64×64. This is resulted from the use of small dataset in experiments and limited availability of training samples prohibit the use of larger inputs. Secondly, classes with limited training samples (see Table 4.4) like Picnic and Festival, filter counts in their convolutional layers must be kept small to achieve successful results. This can be seen in Tables 4.8 and 4.7 respectively. Opposite is true with classes that have more training samples.

Table 4.10: Single class and overall system classification error

Class	Validation Error	Test Error
Graduation	33.33%	44.8%
Selfie	41.82%	29.09%
Festival	27.27%	35.29%
Picnic	47.83%	28.26%
Birthday	32.26%	30.65%
Overall System	36.74%	33.69%

In Table 4.10, we can see that each network achieved relatively successful classification results. Highest classification rate is achieved in the Picnic class with 71.74% accuracy and overall system achieved 66.31% correct classification. Classification rate of Birthday class is marginally below average and improvements in this class will yield better classification rates on the overall system.

4.3 Conclusion

The purpose of this study was to classify social media images in a natural, practical and simple way without limiting their scope. Moreover, we also wanted this classification to allow class expansion without affecting the performance of the previous classes. To that end, convolutional neural networks have been used for classification based on the previous successful results in image classification. In the first part of the study, single networks is used to directly classify complex social media images with fixed classes to show the validity of the approach. Experimental results with 86.27% classification accuracy shows the successful application of this approach on social media images with fixed classes. After the success of this classification example, a modular system is developed to allow class expansion after training. This modular approach uses different networks for the classification of each class. Using this modular system, new classes can be introduced without affecting the performance of the previously trained classes. The results of the experiments on this modular system shows promising results and reveal that the purpose of this study is successfully achieved. For future

work, more complex modular systems can be realized to classify images with multiple classes.

REFERENCES

[1] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[2] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[3] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[4] Dan CireşAn, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012.

[5] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 411–418. Springer, 2013.

[6] Claudio Cusano, Gianluigi Ciocca, and Raimondo Schettini. Image annotation using svm. In *Electronic Imaging 2004*, pages 330–338. International Society for Optics and Photonics, 2003.

[7] James Dowe. Content-based retrieval in multimedia imaging. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 164–167. International Society for Optics and Photonics, 1993.

[8] SL Feng, Raghavan Manmatha, and Victor Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.

[9] Ian Goodfellow, Aaron Courville, and Yoshua Bengio. Deep learning. Book in preparation for MIT Press, 2015.

[10] Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Proceedings of the*

1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition., pages 762–768. IEEE, 1997.

[11] Md Monirul Islam, Dengsheng Zhang, and Guojun Lu. Automatic categorization of image regions using dominant color based vector quantization. In *Computing: Techniques and Applications, 2008. DICTA'08. Digital Image*, pages 191–198. IEEE, 2008.

[12] Md Monirul Islam, Dengsheng Zhang, and Guojun Lu. A geometric method to compute directionality features for texture images. In *2008 IEEE International Conference on Multimedia and Expo*, pages 1521–1524. IEEE, 2008.

[13] I Kanellopoulos and GG Wilkinson. Strategies and best practice for neural network image classification. *International Journal of Remote Sensing*, 18(4):711–725, 1997.

[14] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] Raphaël Marée, Marie Dumont, Pierre Geurts, and Louis Wehenkel. Random subwindows and randomized trees for image retrieval, classification, and annotation. In *Proceedings of the 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) and Sixth European Conference on Computational Biology*, 2007.

[17] Marvin L Minsky and Seymour A Papert. *Perceptrons - Expanded Edition: An Introduction to Computational Geometry.* MIT press Boston, MA:, 1987.

[18] Florent Monay and Daniel Gatica-Perez. Plsa-based image auto-annotation: constraining the latent space. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 348–351. ACM, 2004.

[19] Mark S Nixon and Alberto S Aguado. *Feature extraction & image processing for computer vision.* Academic Press, 2012.

[20] Soo Beom Park, Jae Won Lee, and Sang Kyoon Kim. Content-based image classification using a neural network. *Pattern Recognition Letters*, 25(3):287–300, 2004.

[21] Nikhil Rasiwasia, Nuno Vasconcelos, and Pedro J Moreno. Query by semantic example. In *International Conference on Image and Video Retrieval*, pages 51–60. Springer, 2006.

[22] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for lvcsr. In *Pro2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8614–8618. IEEE, 2013.

[23] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[25] Martin Szummer and Rosalind W Picard. Indoor-outdoor image classification. In *Proceedings of the 1998 IEEE International Workshop on Content-Based Access of Image and Video Database.*, pages 42–51. IEEE, 1998.

[26] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[27] Nai-Chung Yang, Wei-Han Chang, Chung-Ming Kuo, and Tsia-Hsing Li. A fast mpeg-7 dominant color extraction with new similarity measure for image retrieval. *Journal of Visual Communication and Image Representation*, 19(2):92–105, 2008.