Application of deep learning for Urdu numeral recognition

Aamna Bhatti

MSCS00000318704

Supervisor

Dr. Rafia Mumtaz

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree

of Masters in Computer Science (MS CS)

In

School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(August 2021)

Approval

It is certified that the contents and form of the thesis entitled "Application of deep learning for Urdu numeral recognition" submitted by AAMNA BHATTI have been found satisfactory for the requirement of the degree

Advisor: Dr. Rafia Mumtaz

Signature: Rafia Mumtaz

Date: 23-Jun-2021

Committee Member 1:Dr. M. Ali Tahir

Signature: Ali

Date: 23-Jun-2021

Committee Member 1:Dr. Omar Arif

Signature:

Date: 24-Jun-2021

Dedication

Dedicated to my parents, brother, sister, and niece. This thesis is especially dedicated to my boss and mentor Mr. Waqar Khalid without his continuous support and guidance this research wouldn’t have materialized as swiftly.

Certificate of Originality

I hereby declare that this submission is an original work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgment has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation, and linguistics which has been acknowledged.

Author Name: Aamna Bhatti

Student Signature: Aamna Bhatti

Acknowledgment

I am thankful to Allah the almighty to have directed me throughout this research at each and every step and for every new idea which You incepted in my mind to improve. Indeed I could have done very little if it weren’t for Your divine guidance and direction. Whoever helped me throughout the course of my research, whether my family or anyone else was your will, so indeed none be worthy of praise except You.

I am greatly thankful to my beloved parent, siblings, and niece, for continued unconditional support throughout each and every department of my life.

Also, I would like to express special thanks to my supervisor Dr. Rafia Mumtaz for her support and help throughout my research and for encouraging me.

I would also like to pay special thanks to my co-advisors and committee members for their cooperation and tremendous support. Every time I was stuck in some problem, they would come up with a solution. Without their help, I would not have been able to complete my thesis effectively.

Finally, I would like to express my special gratitude to my boss Mr. Waqar Khalid without whom this degree and many achievements wouldn’t have been possible. It was his vision that became a reality and I dedicate this thesis to him entirely.

Table of Contents

List of Abbreviation

DNN	Deep Neural Network
OCR	Optical Character Recognition
DCGAN	Deep Convolutional Generative Adversarial Network
GAN	Generative Adversarial Network
CNN	Convolutional Neural Network
PCA	Principal Component Analysis
t-SNE	t- Stochastic Neighbor Embedding
CBAM	Convolutional Block Attention Module
SE	Squeeze-and-Excitation Network
ResNet	Residual Network

List of Tables

List of Figures

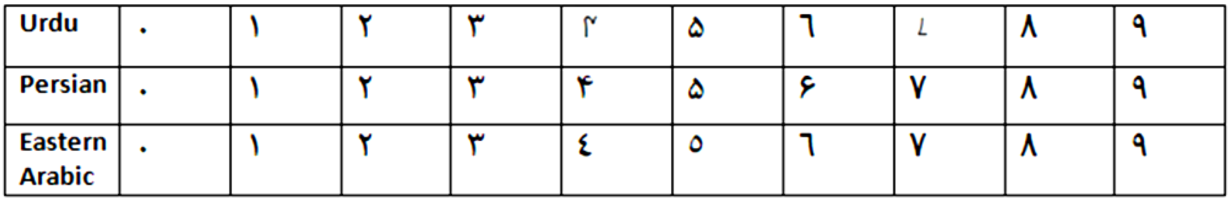
Abstract

Urdu is a complex language widely spoken in South East Asia. Since it is also the national language of Pakistan, Optical Character Recognition (OCR) finds much application of Urdu numerals in our country. In many areas, the Urdu numeral finds its application in Pakistani Currency Notes, Pakistani postage stamps, and automatic dictation. Due to the unavailability of the public datasets Urdu numerals, there isn’t much research done in this domain. While there are many public datasets available for famous languages such as English, Chinese, Arabic, Japanese hence these languages excel in the latest research areas. In this thesis, we provide a novel dataset of Urdu numerals that have been collected keeping in view the dynamics of the real world as every person has a unique style of writing. Since the state-of-the-art techniques in deep learning require bundles of data to train, we have employed the Deep Convolutional Generative Adversarial Networks (DCGAN) which have been rarely explored for this problem. The resulting augmented images have been visualized using t-distributed Stochastic Neighbour Embedding (t-SNE) that further confirms the realness of the images. It is almost impossible to recognize the real and fake images in 2D space. Next, we have to build an Urdu numeral classifier to recognize the diversified Urdu digits. We have employed ResNet18, ResNet18 and Squeeze and Excitation block (SE), and ResNet18 and Convolution Block of Attention Module (CBAM). These modules are tested with and without DCGAN artificially produced augmented data. We conclude that our dataset achieved 100% accuracy on ResNet18 and CBAM model. To further validate this accuracy we have tested the performance of the model using a test set occupied from four sources namely another set of handwritten numerals on the same lines as the original dataset, Pakistani currency notes, numerals written on gadgets with touch and thin strokes using pointer. Our model was able to achieve 95% accuracy on these diversified test sets. Furthermore, this classifier is tested on numerals of Persian, Arabic, and English language. Our model achieved an accuracy of 79.3% on numerals of the Persian language and 54.5% on numerals of the Arabic language. However, the lowest accuracy of 18.4% is achieved on numerals of the English language. These results make our model very reliable to be deployed in any practical application. Using this model can revive our national language and bring it up to speed with the research world.

1 Introduction

1.1. Background

Urdu is one of the cursive languages that is widely spoken and written in the regions of South-East Asia [1]. While it is a national language of Pakistan, it is also very popular in South Asian countries like India, Bangladesh, Bhutan, and Nepal according to the census held in 2001 [2] as well as in the Middle East, Europe, the United States, and Canada. It is the first language of at least 60.5 million speakers, with another 40 million or more who speak it as a second language [3]. The Urdu language has up to 40 letters in the script and 10 numerals. It is an amalgam of many languages hence there are loanwords in the script. Also, Urdu is bidirectional while the script is written from right to left the numerals are written from left to right [1]. Figure 1.1 shows the similarities and differences between the numerals of Urdu language with Persian and Eastern Arabic language.



Urdu	۰	۱	۲	۳	۴	۵	٦	۷	۸	۹
Persian	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
Eastern Arabic	٠	١	٢	٣	٤	٥	٦	٧	٨	٩

Figure 1.1: **Numerals of Urdu, Persian, and Eastern Arabic language**

With the rapid growth of multimedia news and documentation, new challenges are arising in machine learning and pattern recognition [4]. Character recognition is gaining much attention due to advancements in technology such as smartphones and devices for capturing handwriting [5]. Since handwriting is highly writer dependant and every writer has a different style of writing hence there is a need for a highly reliable recognition system that identifies the character input to the application.

In this work problem of classifying Urdu numerals from 0 to 9 is considered which finds uses in various practical applications in finance and administration [6]. They require a remarkable rate of recognition with a minimum error rate. According to Census held in 2017 reports, while only

37.2% of the total population lives in urban areas [7], a system that can recognize Urdu numerals can help teach rural people our national language and thus revive the use of Urdu.

Optical Character Recognition (OCR) is very important research is in character recognition and artificial intelligence [8]. Many applications have been developed using OCR such as automatic license plates, text information extraction, and verification code images [9]. Furthermore, developers working on OCR systems have taken into account a broad variety of features for handwriting digit recognition. Although the majority of features are common, a few of them, such as graph-theoretic methods, gradient-based characteristics, and shadow-based characteristics, use special attributes to boost the classifiers' performance [10].

Machine learning algorithms have revolutionized by providing remarkable results in automatic speech recognition [11], face detection [12], translating text between two languages [13]. They have also proven themselves in advanced applications such as medical diagnosis [14], autonomous driving [15]. Deep learning techniques that are progressing to provide state-of-the-art methods have surpassed human performance [16]. But the need for the data is always there to produce results. Furthermore, most of the deep learning methods require a large amount of data to train even to solve a single task.

The scarcity of data is the bottleneck for any machine learning algorithm and to get around the matter of scarce training data, data augmentation is most commonly used. Data augmentation is a technique that is used to regularize models. Synthetic images are created by applying numerous transformations to the original dataset [17]. It generates artificial data from the available data. Numerous photometric and geometric transformations are applied on original images such as rotation, translation, flipping, and addition of noise creates new images. Because of the random aspect of data augmentation, it has the ability to produce ‘unlimited’ data by supplementing existing data. While this is a viable solution to the problem of insufficient training data, it is not widely accepted since the form of data augmentation and its requirements must be tailored to the task at hand. Also, the type of transformations that are needed to be applied depends on the problem we are trying to solve. In certain cases, some transformations change the class of the image for example rotating ‘6’ 180 degrees change 6 to 9 and thus cannot be assigned the same class. Furthermore, the parameters for augmenting data introduce a new set of important hyperparameters that have a significant effect on the error produced by the deep learning algorithms [17].

Although the most popular method of data augmentation is to randomly augment data using basic transformations, more complex methods for synthesizing extra training data have also been suggested. However, since the distributions of actual and synthetic data are usually different, this is a difficult challenge that necessitates further post-processing on the synthetic data. The non-linearity in the real world is more complex to be modeled by mere rotation and translation of the images. As a result, using a generative model to augment data will boost the training required for deep learning approaches. This generative model is normally fine-tuned or matched to fit the actual data distribution for rendering-based approaches.

Generative Adversarial Networks (GANs), recently proposed by Goodfellow et al. in 2014, provide an appealing way of automatically learning a generative model by simply training a typical deep neural network. A GAN is made up of two sub-networks that compete with each other: the generator and the discriminator. The generator creates data from a noise vector as input. The discriminator is a regular classification network that takes input data from the generator as well as real data. The discriminator's task is to correctly identify each image given input as real or synthetic, while the generator's goal is to generate images that look like real data in order to trick the discriminator. GANs have shown promising results in activities such as image generation [18][19], synthetic data generation [20], and domain transfer [21]. However, though GANs produce amazing results when trained on massive data, how GANs behave when trained on a small amount of data is still a subject of active research.

One of the most common Deep Neural Network (DNN) learning algorithms that can perform classification tasks directly from images is the Convolutional Neural Network (CNN). It automatically generates features using convolution layers. And then the probabilities of each class are predicted by connecting a fully connected layer. However, this revolutionary method is prone to poor performance and overfitting because in classification some of the features at high frequencies are not useful. Inspired by how humans visualize scenes by focusing more on important details while suppressing unimportant attributes, attention mechanism is incorporated in CNNs. This method improves the representation ability of CNN and focuses on important information. Convolutional layers extract important features by combining spatial and channel information. By applying attention modules each of the layers can learn not only ‘what’ but also ‘where’, to focus on spatial and channel axis. This helps in information flow in the model by learning which information to suppress and which to emphasize [22].

For this thesis, we will focus on data augmentation methods in the context of image classification of Urdu numerals. Image classification is the task of categorizing each image by assigning a label. Each numeral belongs to the class from 0 to 9. Figure 1.2 illustrates an Urdu numeral and its classes.

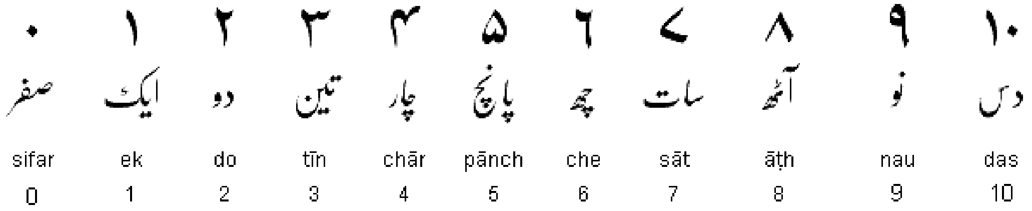

۰ ۱ ۲ ۳ ۴ ۵ ٦ ۷ ۸ ۹ ۱۰
صفر ایک دو تین چار پانچ چھ سات آٹھ نو دس
sifar ek do tīn chār pānch che sāt āṭh nau das
0 1 2 3 4 5 6 7 8 9 10

Figure 1.2: Classification of Urdu numerals

1.2. Motivation

The motivation behind this topic is that several automatic systems are available in the market for offline handwriting recognition for few major world scripts, such as English, Chinese, Arabic, Japanese, etc due to the abundance of public data available for these languages. While Urdu which is very popular in South-Asian countries has been neglected for so long. One reason is the lack of public datasets available to bring the Urdu language to a competitive level as data is the essence of machine learning programs. So the main motivation behind developing a handwritten Urdu numeral classifier is that majority of the work has been done on English numerals so it is the need of the hour that techniques of deep learning are applied to our mother language ‘Urdu’. In this way, the problems related to Urdu character recognition could be solved as efficiently as they are solved in other languages.

1.3. Problem Statement

Since Urdu numerals have a low technical exposure, the aim is to get Urdu up to par with other languages. Since the dataset contain only 9800 images so the proposed idea is to apply GAN on a novel dataset that is collected on lines of MNIST to increase the size and then see how the CNN performs in terms of accuracy when augmented data is added. The aim is to add attention mechanisms in the standard ConvNet that are currently available.

There are two main objectives in the execution of the idea. The first is to apply GAN to augment the data as every person has a different writing style and strokes, augmentation can help incorporating non-linearities or dynamics of the real world. This would be tested with different variations of GAN available. The second is applying CNN on the data to achieve the recognition task and see which of the network gives the best results in terms of accuracy.

1.4. Contribution of the Thesis

The contributions of this thesis are:

1. DCGAN is applied to augment the Urdu numerals dataset. The results are further enhanced by rigorous experimentation by employing traditional ways of augmentation on the dataset, which has not been explored earlier on this problem.
2. The outcome of the experiments is evaluated using: 1) PCA 2) t-SNE. It is observed that artificial images form tight clusters with original images in high dimensional space.
3. The state-of-the-art Urdu numeral classifier is developed by training extended data using an attention mechanism in Convolutional Neural Network. The classifier achieved an accuracy of 100%.
4. The results are further validated on four test sets and numerals of three languages.

1.5. Organization of Thesis

This thesis consists of seven chapters. It is organized as follows:

1. Chapter 1 introduces Urdu Numeral and the role of OCR, motivation, problem statement, the contribution of thesis in the field of Urdu Numeral Recognition.

2. Chapter 2 provides a literature review on digit classification of Urdu language and other different existing works and techniques available that has been implemented for similar problems. The focus is on the work done for recognition of Urdu numeral, GAN for data augmentation, and attention mechanism in CNN

3. Chapter 3 discusses the proposed method of implementation for the task at hand. Also, the discussion is provided for existing datasets and proposed dataset that is obtained to efficiently solve the problem at hand. An elaborated discussion of the Deep Convolutional generative Adversarial Networks (DCGAN) and attention mechanism in Convolutional Neural networks have also been provided in this chapter.

4. Chapter 4 presents results and discussions. It explains evaluation metrics, results, and performance comparisons between all networks.

5. Chapter 5 is the conclusion where a summary of the thesis is provided along with problems encountered and suggestions for future work.

2. Literature Review

Numeral Recognition has been around in computer science for a few decades now. However, deep learning techniques have not been extensively used to solve this problem. This thesis focuses on applying the latest deep learning research to classify the images of digits. The literature review on Urdu Numeral Recognition is discussed in detail in this chapter. The work done is divided into three parts as (1) Urdu Numeral Recognition (2) Data Augmentation using GAN (3) Attention Mechanism in CNN.

2.1 Urdu Numeral Recognition

Due to the abundance of public data available for English, Arabic, Chinese, Japanese languages many offline systems for handwriting recognition are available. In [23] techniques of analysis-by-synthesis origin are used for character recognition. Syntactic approaches to character recognition are built on this foundation. With the developments in the field of deep learning, Urdu handwriting recognition has advanced to a new level. Urdu text has been extensively analyzed from every perspective, from CNN's to BLSTMs, and ultimately to autoencoders. In [24] used stacked autoencoders to extract features from pixel values of 178570 ligatures. They achieved 96% accuracy using 3732 class images. [25] used CNN to test 18,000 Urdu ligatures from 98 different groups and achieved a classification rate of up to 95%. With 99.02% accuracy, SVM was used to correctly identify Urdu ligature components [26]. [27] address developments in preprocessing methods for input data ranging from plain handwritten documents to deformed images, as well as images of perspective variance and context clutter. They argue that using one preprocessing technique is not enough to achieve high accuracy; however, a combination of preprocessing techniques must be used to produce a good result. There are essentially two kinds of recognition systems: online and offline [28], [29]. When a person writes on a special writing pad or surface, the machine identifies it and translates it into codes. Offline recognition systems use images or documents as input and are then translated to digital form. Offline identification is done in stages, with images being segmented, cropped, resized, and features removed before being identified [30]. The dissertation on Urdu numerals using offline recognition is presented in this article [30]. A recognition method is suggested in [31] based on feature extraction by segmentation and slant elimination by slant analysis and search dictionary for classification. After applying bidirectional long short-term memory (BLSTM) networks in 1-D to 700 text lines (including Urdu handwritten samples and Urdu numerals) in [32] obtained a 6.04–7.93% error rate. In [33] use Kohonen Self Organisation Maps on a total of 6000 Urdu numerals and achieve a 91% quality. A paper similar to this is in [34], in which Urdu text is combined with (MNIST) to learn similar trends. By pre-training the network on the Modified National Institute of Standards and Technology dataset (MNIST), they used CNN and multi-dimensional long short-term memory (MDLSTM) on UNHD samples.

Work done in other languages play a very important role to solve the task in our language. For instance [35] describes work done on Eastern Arabic numerals using OCR in 2015. [36] uses Arabic Handwritten Digits Database (ADBase) to equate the problems with Latin and Arabic handwritten numerals in [37]. [38] used a genetic algorithm to extract local features from a dataset of handwritten Bangla digits, which were then fed into an SVM. It yielded 96.7% promising results. Finally, MetaQNN [39], which was proposed in 2018, is a recent noteworthy technique. The creation of CNN architectures is focused on reinforcement learning, and it has origins in the neuro-evolution of CNN committees. When using an ensemble of the most effective discovered neural networks, it has an error rate of 0.44% and 0.3%, respectively.

2.2 Data Augmentation using GAN

Due to the unavailability of the public dataset, online or offline data augmentation has not been yet applied to the problem of Urdu digit classification. Also, there is no work cited to the best of our knowledge of the implementation of GAN in this area. Hence work done in other languages is presented that will help us solve the task which is very similar to the idea presented in this paper. [40] generated handwritten Arabic digits in the numeral script of Eastern Arabic. They used different variants of GAN such as deep convolution GAN (DCGAN), Bidirectional GANs, (BiGANs), VanillaGANs, and WassersteinGANs (WGANs). With accuracy values of 96.815% and 69.93%, respectively, DCGAN outperforms the other GANs in the native-Arabic human benchmark. DCGAN was applied on the three popular Bangla handwritten datasets Bangla Lekha-Isolated, CMATERdb, ISI, and their own dataset Ekush in [41]. They said that DCGAN produced Bangla digits successfully, making it a reliable model for generating Bangla handwritten digits from random noise. To improve the efficiency of the classifier, [42] used

GAN to increase the data. They researched how much artificial images can be produced before output begins to deteriorate. They tested the idea on four datasets of handwritten digits which were Latin, MNIST, Latin, Devanagri, Bangla, and Oriya. Though the performance improved for a bit, after a while, too many GAN-produced images caused the performance to deteriorate. [43] claims that the traditional data augmentation method may lead the generator to incorrectly learn the distribution of the augmented data, which may vary from the original data distribution. They then suggest a principled system Data Augmentation Optimized for GAN (DAG), to allow the use of augmented data in GAN training to enhance the original distribution's learning that minimizes the divergence between the original and model distribution. They applied DAG to different GAN models such as conditional and unconditional GAN, CycleGAN, and self-supervised GAN using datasets of a medical and natural image. Their result showed that DAG achieved consistent and significant improvements in these models.

2.3 Attention Mechanism in CNN

The convolutional block attention module (CBAM) is a recently introduced module in deep learning. Since the attention mechanism has never been applied in numeral recognition of any language to the best of our knowledge, hence we have considered the research in other areas. It has been widely applied in natural language processing tasks. In [44] a large long short-term memory architecture (LSTM) performs better on limited vocabulary using the attention mechanism. [45] establishes a differentiable attention mechanism that emphasizes the neural network to focus more on different parts of the given input. For visual tasks like image, classification attention has brought about revolutionary changes in the visual recognition world. The spatial attention component is modified in [46] to generate different spatial attention maps customized for diversified multi-label learning. The enhanced local features obtained as a result of this are aggregated into non-local representation and then they are applied in CNNs to get remarkable accuracies. Components from Inception and ResNet architecture are combined with visual attention networks to achieve exceptional performance on the ImageNet dataset [47]. Squeeze and excitation block investigates the relationship between channel-wise features. Like attention, it works on channel-wise feature responses which are recalibrated to model inter-dependencies [48].

After reading the state-of-the-artwork in this field following table summarizes the literature review.

Table 2.1: Summarization of Relevant Techniques

Paper	Technique	Dataset	Result and Limitation
Urdu Numeral Recognition			
[2]	Convolution Neural Network	8000 Urdu numeral	Character recognition of rotated, mirror-text, and noisy images
[33]	Kohonen Self Organisation Maps	6000 Urdu numerals	91% quality
[39]	Different Daubechies Wavelet for extraction and. Multilayer Neural Network has been used for classification.	2000 Urdu numerals	Accuracy of 92.07%
[15]	Deep auto-encoder and convolutional neural network	Written by 900 individuals	Accuracy of 97% for digits
Generative Adversarial Network			
[40]	a Basic GAN, deep convolution GAN (DCGAN), Bidirectional GANs, (BiGANs), Vanilla GANs, and Wasserstein GANs (WGANs).	291 images of Arabic Dataset	Accuracy values of 96.815%.
[41]	DCGAN (Deep convolutional generative adversarial networks)	Three datasets: CMATERdb, BanglaLekha-Isolated, ISI, and	Successful generation of Bangla digits

		their own dataset Ekush	
Attention Mechanism for image classification			
[46]	Global spatial attention mechanism in CNNs	Medical images	Accuracy of 84.8%.

We can observe after reviewing the state-of-the-art techniques that have been proposed in recent years that Urdu numeral generation and recognition are unexplored while the digits of various languages have been experimented on using the latest research. In this thesis, we aim to apply a deep convolutional Generative Adversarial Network for image generation and then building an Urdu numeral recognizer by incorporating an attention mechanism on a dataset that also contains DCGAN augmented images.

3. Methodology

Urdu Numeral Recognition is a relatively unexplored area. In this thesis, we provide Urdu numeral recognition and discus every step in detail in the following paragraphs. The methodology for Urdu numeral recognition is given below in Figure 3.1. This chapter is divided into three sections. The first section discusses the existing dataset and the reason for gathering this new dataset of Urdu numerals. We then provide a detailed explanation of the image acquisition procedure that we followed and the pre-processing steps that we have taken to add diversity and dynamics to the real world. The second section provides the description of the Generative Adversarial Network and our proposed approach to apply deep convolution Generative Adversarial Network. We have discussed the generator, discriminator, and loss function used as part of our approach. In the last section, we have demonstrated the Convolution Neural Network and our proposed architecture for the classifier. We have given an in-depth insight into various models that we have used to obtain the best classifier.

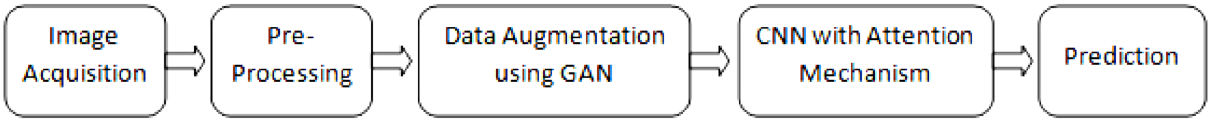


Figure 3.1: Methodology Block Diagram

3.1. Dataset

3.1.1. Existing Datasets

Once we have reviewed the state of art. The next step in deep learning for a problem related to OCR is the dataset. Data used in any machine learning algorithm needs to be standardized to get exemplary outputs. While defining the problem statement no dataset meeting our needs were found. There are datasets comprising Urdu text and spoken words by EMILLE (Enabling Minority Language Engineering) [51] containing 67 million South Asian words. Urdu Printed Text Image Database (UPTI) [52] was created by Sabbour and Shafait which contained 10,063 text lines artificially generated. On similar lines Centre for Language Engineering (CLE) [53] has extracted 192 million words from various domains such as news, sports, culture, finance, and consumer information. Urdu Nastalique Handwritten Dataset (UNHD) was created by Ahmed in 2013 [54] that comprises of total words of 312,000 in 10,000 text lines. In all the available datasets for the Urdu language either there were no numerals or very few numerals were found which were not enough to build and train the model.

Table 3.1: Existing Dataset

Dataset	Content	Size
EMILLE (Enabling Minority Language Engineering).	South Asian words	67 million words
Urdu Printed Text Image Database (UPTI)	artificially generated text lines	Total 10,063 lines
Centre for Language Engineering (CLE)	From domains such as news, sports, culture, finance, and consumer information.	192 million words
Urdu Nastalique Handwritten Dataset (UNHD)	Urdu words	Total 312,000 words in 10,000 text lines.

3.1.2. Proposed Dataset

We sought to build our own dataset to continue with the research work as current datasets did not match the requirement of our research

3.1.2.1. Image Acquisition

Our dataset consisted of a total of 9800 images of Urdu Numeral categorized in 10 classes from 0-9. We gathered the data from 200 different people. They belonged to different age groups from 4 to 81 years. We consciously did this to add versatility to our dataset since each person has a different style of writing. The aim was to add the dynamics of the real world so that when this model trained on this dataset is deployed in the real world can perform exemplary. Numerals were written on white paper with a black marker, and inspiration was gathered from the

Modified National Institute of Standards and Technology (MNIST) and Arabic Handwritten Dataset (AHD).

3.1.2.2. Pre-Processing

We took pictures of the numerals from the CAM Scanner application. Once we had the images of A4 paper with numerals written on it, we then cropped them using the connected component labeling technique. Connected component labeling or connected component analysis is an application of graph theory. It is used to determine the connectivity of regions that resemble blob in the binary image. Hence the first step was to convert it into binary images so that the pixels have either 0 or 255 values i.e. either they are considered as background or foreground. The binary image obtained often contains noise at the background these are the unwanted the pixel that need not be included in the images of numerals. On the binary image, we compute the boundary box which provides us the starting coordinate value x and y and the width and height of the connected component. Using the bounding box information we crop the region and then save it as a separate image of 32 ×32 sizes. To remove noise we filter out the bounding box that is either too small or too large. The process of connected component labeling is shown in Figure 3.2. In Figure 3.2, an image is converted into a binary image and a connected component is identified using a bounding box. Figure b further highlights the connected component and Figure c shows the cropped version of numeral 4.

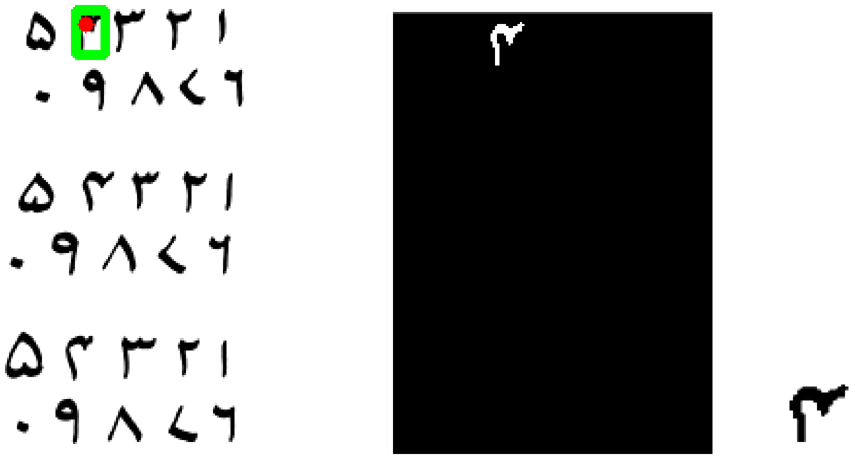


Figure 3.2: Connected Component Labeling (a) Original Image. (b) Binary Image. (c) Cropped Numeral

We subconsciously wrote digits belonging to the same class on one paper so that once we have the entire cropped image from one paper we could place them all in the same folder and label the folder with a class name. This would provide us with labeled data at the end.

To add more variety to the dataset we add some hard cases as well. We also gathered the data that was written on crumbled pages. We later added dots on the page so that our dataset contains the dynamics real world which is highly non-linear as people don’t always write on clean pages. These modified pages were filtered using a Gaussian filter which minimized the effect of noise. We tested different sigma values and found 3 to be the most ideal. At sigma value 3 produced the most optimal results. The original image of hard case data and the filtered image is shown below in Figure 3.3.

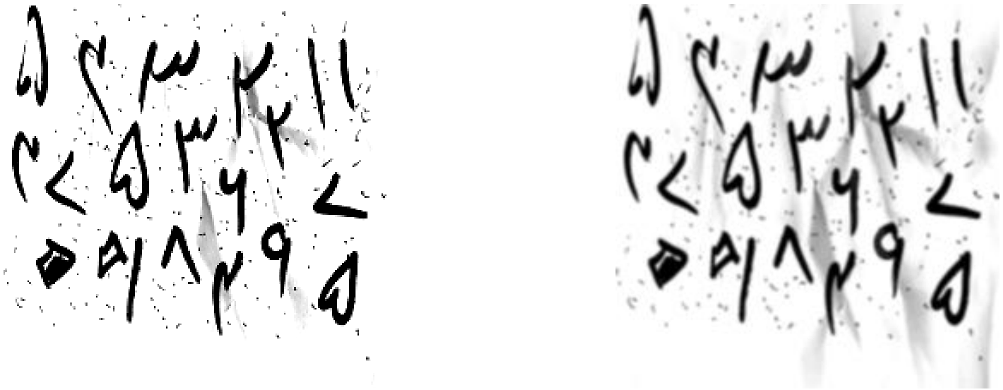


Figure 3.3: Hard Cases (a) Original Image. (b) Gaussian Filtered Image

Then we threshold the image to get a binary image so that we only have background white and foreground black including the noise. We performed experiments with different threshold values as can be seen below in Figure 3.4. After a repeated experiment with the threshold values, we observed that if the threshold is set to a higher value i.e. 150 the numerals appear very sharp but so is the noise as can be seen in Figure 3.4a. If we subdue the noise by keeping the threshold low i.e. 50 the edges of the numerals in images also blurs and we lose some pixels to the threshold as can be observed from Figure 3.4b. But with the threshold value set to 120 most optimal result was obtained shown in Figure 3.4c.

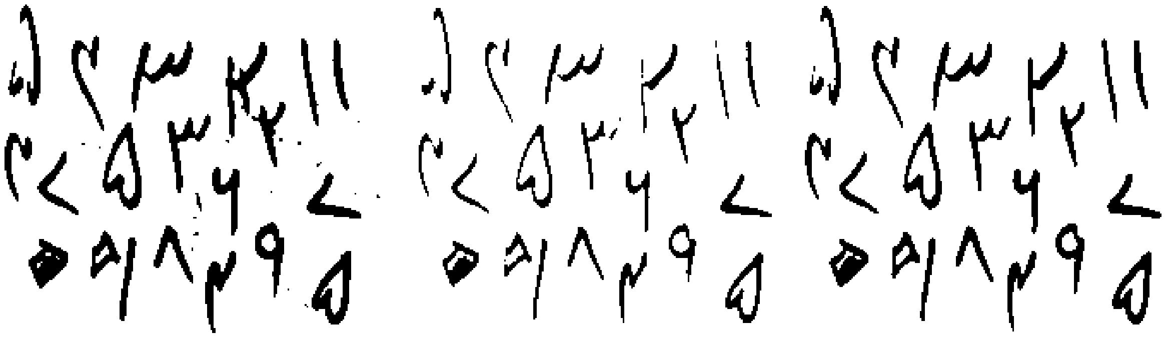


Figure 3.4: Different Threshold Value (a).High Threshold Value 150 (b) Low Threshold Value 50 (c) Optimal Threshold Value 120

3.2. Generative Adversarial Network

Deep neural networks have been around for a long time but as the network gets deeper the more data is required to train it. Since their inception, Generative Adversarial Networks (GANs) have been particularly effective in the generation of the image. It consists of two networks i.e. generator (G) and discriminator (D) playing a min-max game. The G takes as input random variables from a latent space and output the generated images labeled as 0. These generated images are mixed with real images labeled as 1. Collectively they form a dataset that serves as input for D. Discriminator is designed to perform binary classification i.e. classify images as real or fake. While G aims to maximize the error of generated images, D attempts to reduce the errors for fake and real images. This can be expressed by the following equation:

$$min_G max_D \, V(G,D) = \, E_{x\sim p_{data}\,(x)}[\,[\log D(x)\,] + \, E_{z\sim p_x(z)}[1 - \log D\big(G(z)\big)]\,] \tag{3.1}$$

x represents an example of an image example, p is the distribution of data, and latent variable is represented by z.

In this thesis, we have proposed the use of deep convolutional GAN to augment Urdu numerals. To enhance the results and inculcate the different writing styles, we have experimented using traditional augmentation techniques in the dataset. Moreover, we have compared the results of DCGAN when no augmentation is applied on the dataset and when the dataset is augmented using traditional ways before providing them as input to DCGAN. These experiments helped improve the realness of the augmented images generated by DCGAN. We have also evaluated the results of DCGAN using t-SNE.

3.2.1. Proposed Approach for DCGAN

We propose the use of Deep Convolutional Generative Adversarial Networks (DCGAN) to augment the Urdu numeral dataset. The architecture is represented in Figure 3.5

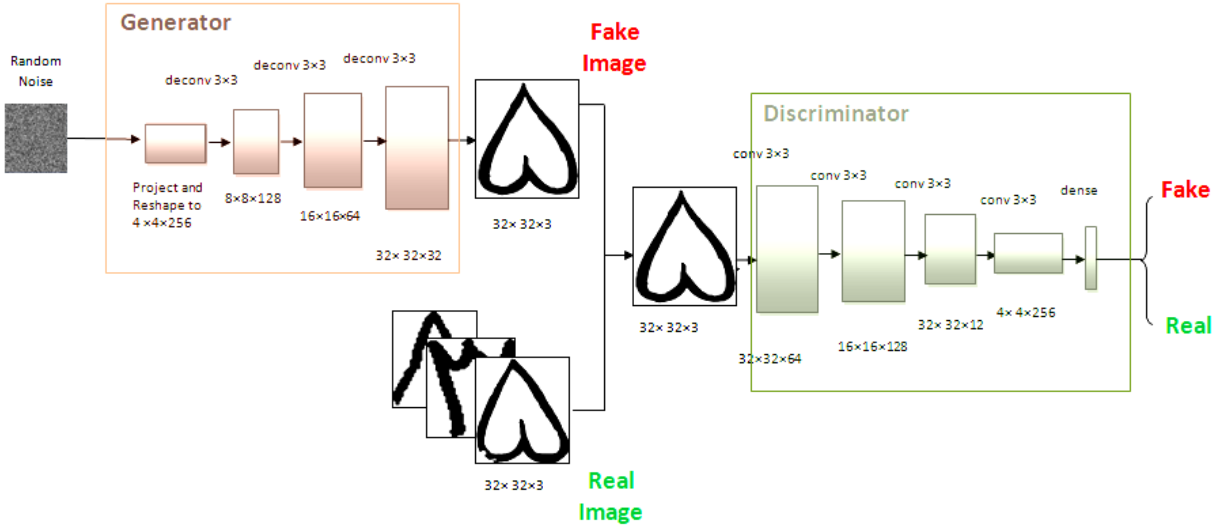


Figure 3.5: DCGAN Architecture for Urdu Numerals

3.2.1.1. Generator Network

The purpose of the generator network here is to create new and fake but realistic handwritten digits. It does this by taking as input random 100 × 1 noise vector which is then provided to dense layer to obtain 128 different 7×7 feature maps. Then there are three convolution transpose layers to upsample the obtained representation with ReLU activation function in between except for the last convolution layer. This allows the model to quickly learn saturation and cover the color space of the training distribution [56]. In our proposed architecture using three convolution transpose layers, we have upsampled the representation of size 4 × 4 × 256 to an image of size 32 × 32.

3.2.1.2. Discriminator Network

The aim of the discriminator network is to determine whether the images are fake or real. The network takes a combination of 32 × 32 size originals images from the dataset and images generated by the generator as input followed by three convolution layers with leaky ReLU activation function in between as recommended by Radford et al. [57]. The last convolution layer uses a sigmoid activation to determine whether the image was original (real) or generated by a generator (fake).

3.2.1.3. Loss Function

In this thesis, we have used the min-max loss function given by Equation 3.1. Binary classification is employed in discriminator networks as it needs to distinguish between real and fake images. Binary Cross-Entropy (BCE) is given as:

$$J_{BCE}(\Theta) = \frac{1}{M}\sum_{m=1}^{M}[\, y_m \, log \, (\, h_\theta \, (x_m)) + (1 - \, y_m) \, log \, (1 - \, h_\theta(x_m))] \qquad 3.2$$

M here represents training samples in a mini-batch, y_m is the label of a target for training sample m (for real image label is 1 and for fake image, the label is 0), the input for a training sample is given by x_m whereas h_θ is model with network weights ϴ. We are summing over variable M as shown by summation at the start of Equation 3.2. This gives us the average cost of all examples in the entire batch. Moreover, if 1 is the output of the model then the loss will be –log (1) = 0 and, hence training sample is a real image. On similar lines $(1-y_m)$ log $(1-h_\theta(x_m))$

3.3. Convolution Neural Network

In the field of computer vision, fine-grained image recognition is a significant subset [60][61][62]. To maintain “big variations within the class and minor differences between classes." makes it a difficult problem. Fine-grained image classification has more research value and practical importance as compared to conventional image classification. It requires learning the subtle features thoroughly while placing local visual distinctions with appropriate discrimination.

Convolution Neural Network consists of the following major components.

- **Convolutional Layer:** This layer detects the features in an image or feature map given at its input using a filter kernel. Filter kernel slides over the image and computes the product between features and the portion that is being scanned by the kernel.
- **Pooling:** This operation is responsible for reducing the size of the image or feature map while preserving the important characters in the image. This helps to improve the efficiency of the network and eliminates over-learning. The most commonly used pooling is Max-Pool that preserves only the maximum value in window

- **Activation Function:** It provides non-linearity to the network. Usually employed after convolution and fully connected layer. Most commonly used activation function is Rectified Linear Units (ReLU) that replaces the negative value in feature map with 0 and pass the positive values as it is. The ReLU function is defined by max operation i.e. *ReLU(x)=max(0,x),* demonstrated by Figure

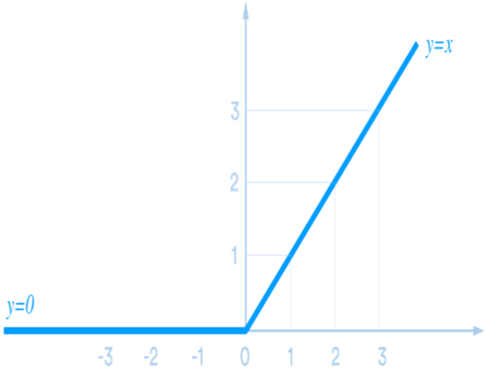


Figure 3.6: Rectified Linear Units (ReLU)

- **Fully connected layer:** This layer is usually employed at the end of the network. Many researchers do not consider it as part of the CNN model. It receives input vector and applies linear combination and activation function to input values and generate output vector.

Urdu numerals provide sufficient complexity. Unlike other languages, Urdu digits are very similar to one another as can be seen in Figure 3.7.

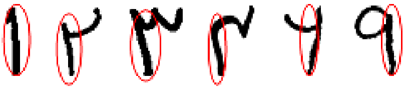
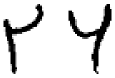
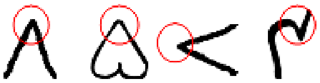
Type of structural features	Urdu Numeral
Number of horizontal lines	
Mirror image	
Cusp and its position	

Figure 3.7: Structural Features in Urdu Numerals

While numeral 7 is at 45 degrees to digit 8 which in turn is similar to 5. Also, a total of six digits contain horizontal lines. Digits 2 and 6 are mirror images of each other while the difference between 2 and 3 is of an additional arc.

Dealing with such images requires concentrating on features that distinguish them from the rest of the classes. Inspired by how humans visualize scenes by focusing more on important details while suppressing unimportant attributes, attention mechanism is incorporated in CNNs. This method improves the representation ability of CNN and focuses on important information. In this thesis, we present a handwritten recognizer for Urdu Numerals using Convolution Based Attention Module and Squeeze-and-Excitation Network (SE) with ResNet18. These are then compared with vanilla ResNet18 to compare the differences in accuracies.

3.3.1. Proposed Approach for Convolution Neural Network

In order to perform the classification of handwritten Urdu numerals, we used 3 different variations of ResNet architectures i.e. vanilla ResNet18, a combination of ResNet18 and SE, and finally the combination of ResNet18 and CBAM. Moreover, we trained and tested these variations with and without GAN augmented images. Hence total of six experiments was conducted to get a comparison for the best results.

3.3.1.1. ResNet Architecture

To get an understanding of the performance of the base model, we applied ResNet18 which consist of 18 weighted layers, with shortcut connection in between each pair of 3×3 filters.

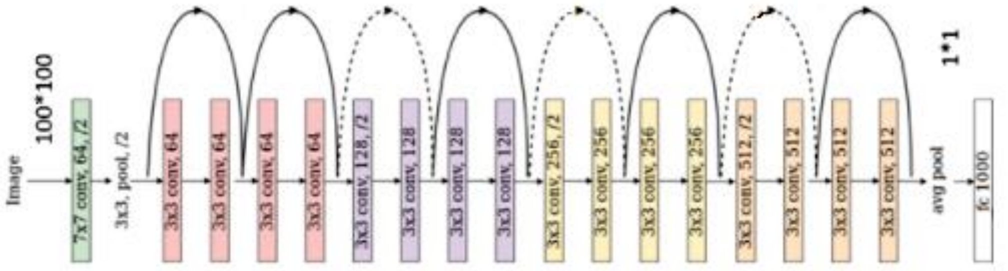


Figure 3.8: ResNet18 Architecture [66]

3.3.1.2. Squeeze-and-Excitation Module (SE)

A new architectural unit was introduced in 2019 - the Squeeze-and Excitation (SE) block which was the foundation block for the winner of the ILSVRC 2017 challenge [63]. It focuses on channel-wise features and gives a higher preference to specific channels over others. This is done by scaling more important channels by a higher value.

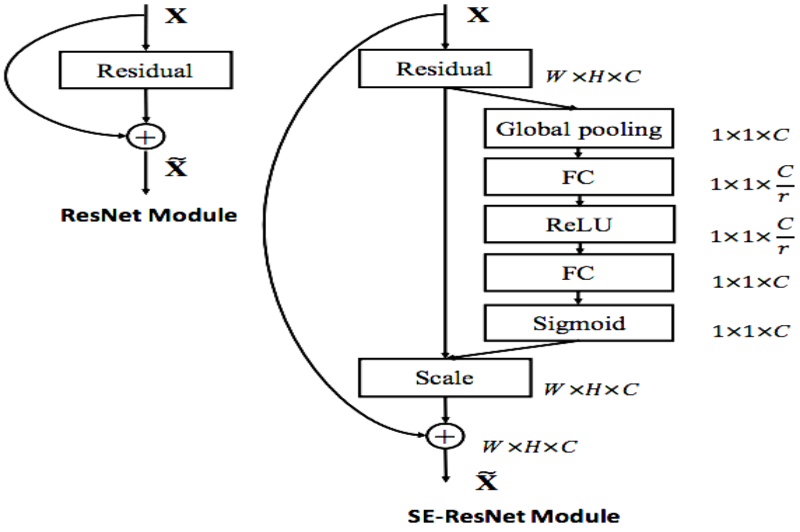


Figure 3.9: Squeeze and Excitation Block in ResNet [67]

The Squeeze and Excitation block is shown in Figure 3.9. SE block improves the power of representation in a network by enabling the feature recalibration in channels. The convolutional block is given to it as input. It then squeezed each channel in one numeric value by taking average pooling. Non-linearity is added by ReLU following dense layer and hence output complexity in the channel is reduced in the ratio. Then another dense layer is added following the sigmoid that gives a smoothing effect to each channel. Finally, each feature map in the convolutional block is weighted based on the excitation.

3.3.1.3. Convolutional Block Attention Module (CBAM)

Convolutional Block Attention Module (CBAM) module was presented in 2018 with the aim of boosting the representation power of CNNs [65]. It makes a CNN focus on important foreground information and gives less preference to useless background information. For instance, for object detection target class of objects is the important information based on which the model should work. For image classification, the background noise caught during the image acquisition process is dealt with as unnecessary information. CBAM is usually applied to spatial and channel dimensions.

3.3.1.3.1. Channel Attention

In this thesis, we have first applied the channel attention module and then the spatial attention is applied across spatial dimensions. Finally, the result is added to the previous convolutional layer. This arrangement of attention modules is of the same order which was used in the original CBAM paper [65].

Given a feature map F ∈ R with dimensions C×H×W as input. 1D channel attention map $Mc \in R^{C\times1\times1}$ and 2D spatial attention map $Ms \in R^{1\times H\times W}$ are sequentially inferred by CBAM. The complete process is summarized by the following equation as given in [65]:

$$F^{'} = M_c(F) \otimes F \qquad 3.3$$

$$F'^{'} = M_s(F') \otimes F' \qquad 3.4$$

⊗ represents element-wise multiplication. F" denotes final refined output. Figure 3.10 represents the CBAM module

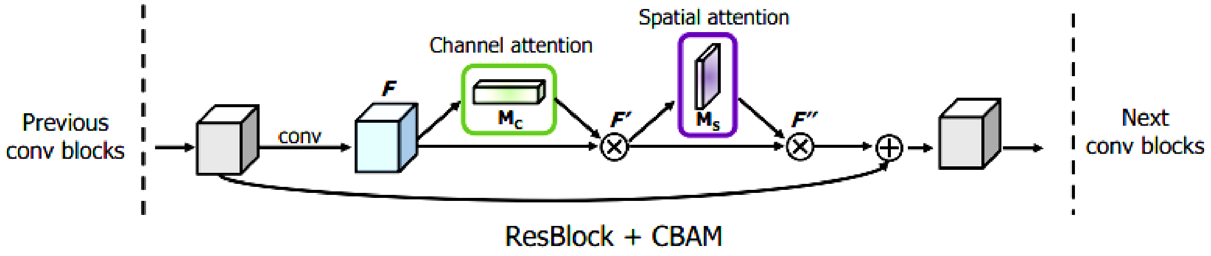


Figure 3.10: Convolutional Block Attention Module (CBAM) [65]

In our approach, we first applied channel attention which explores the relationship of features between inter-channels. Given an image, it concentrates on 'what' is important by squeezing the input feature map. Features are both max-pooled and average-pooled simultaneously. This improves the power of representation in networks as compared to using each independently. The operation of channel attention is provided by following Equation 3.5 in [65].

$$M_c(F) = \sigma(W1\left(W2\left(F_c^{avg}\right)\right) + W1(W0(F_c^{max})) \qquad 3.5$$

3.3.1.3.2. Spatial Attention

We then employed spatial attention that utilizes the relationship of features at the inter-spatial level. Unlike channel attention, it concentrates on 'where' information by first average pooling and max pooling features along the channel axis and then concatenating them to generate an effective feature descriptor. This highlights the informative region. Equation 3.6 demonstrates the spatial attention mechanism

$$M_s(F) = \sigma\left(f^{7\times7}([AvgPool(F); MaxPool(F)])\right) \qquad 3.6$$

$$= \sigma\left(f^{7\times7}\left(\left[F_s^{avg};\ F_s^{max}\right]\right)\right)\ M_s(F) \qquad 3.7$$

4. Results and Discussion

4.1. Deep Convolutional Generative Adversarial Network

To experiment with DCGAN we provided the dataset of Urdu Numeral containing 9800 images of classes 0 to 9. The sample of the dataset is given in Figure 4.1

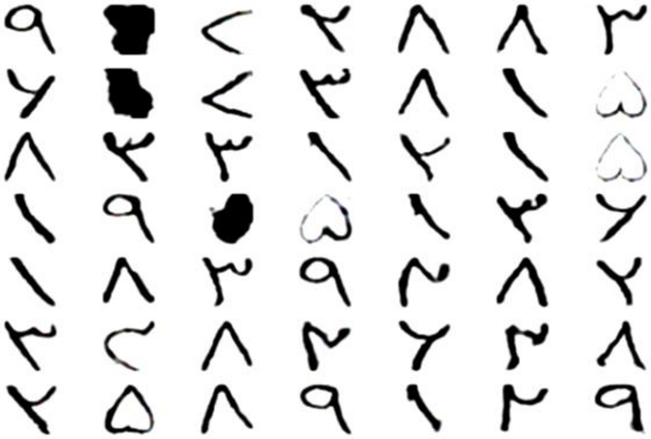


Figure 4.1: Sample from Original Dataset

4.1.1. Experimentation and Results

We trained the generator and discriminator network using a learning rate of 0.0002 and momentum of 0.5. The number of epochs was 200 using a mini-batch of 128. Initially, we trained the DCGAN for 50 epochs and the images produce resembled numerals. The quality of images though further improved after 200 epochs but some numerals were still unrecognizable as can be seen in Figure 4.2a. Then we augmented the dataset using traditional ways and provided the increased dataset to the DCGAN and observed the results. The dataset was augmented keeping in view the nature of the numerals because some pairs of numerals such as 2 6 and 7 8 are very much similar and mere rotation can change their class instead of producing an augmented image. Another key point that was kept in view was that the result of DCGAN without augmentation showed that some numerals produced are more real than the others such as DCGAN was able to produce 3,8 and 9 better than the rest. So our focus was mainly on those numerals that were hard for the DCGAN to generate. Hence numerals 0 and 5 were flipped and added to the dataset. Digits 1, 2, 3, 6, and 4 were randomly rotated between 0 and 10 degrees. Numeral 8 was rotated anticlockwise for 45 degrees and was added to numeral 7. After these operations, the size of the dataset was increased from 9800 to 22000. We analyzed that after now after 150 epochs the results were very real. The results of DCGAN with and with data augmentation can be seen in Figure 4.2b. Hence the augmentation of few classes refined the results altogether and due to the generation of new images, the data gets more supplemented and we get clearer margins between classes.

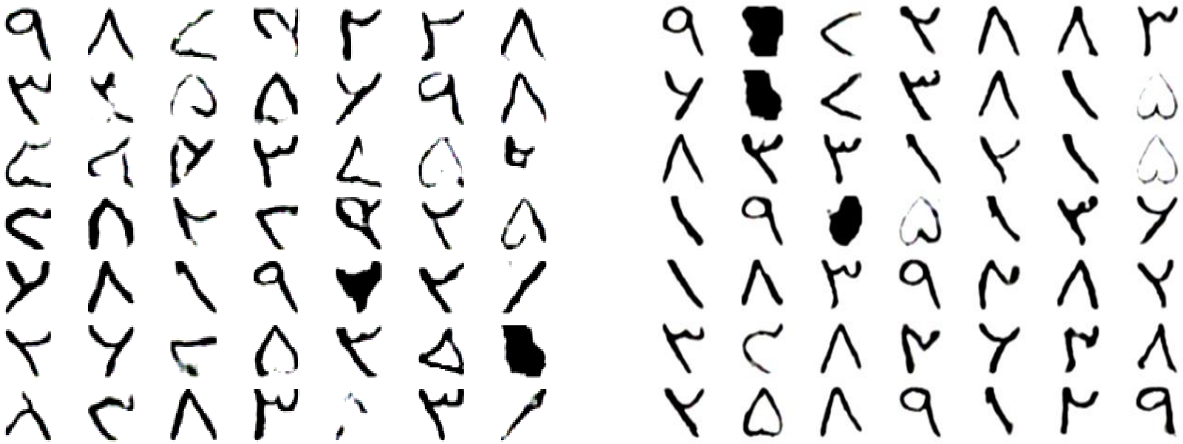


Figure 4.2: DCGAN generated Images (a) Without Augmentation (b) With Augmentation

It is very evident from Figure 4.2b that it is hard to identify the fakeness in the numerals generated by the DCGAN. Numerals 0, 1, 2, 4, 5, 6, and 7 are as real as the original dataset.

We then plotted the accuracy and loss graph during generator and discriminator training as shown in Figure 4.3. It can be seen that the loss of the generator and discriminator for both fake and real images is almost 0.5, and the accuracy of the discriminator network is around 80% which indicates that the model has converged to a stable equilibrium.

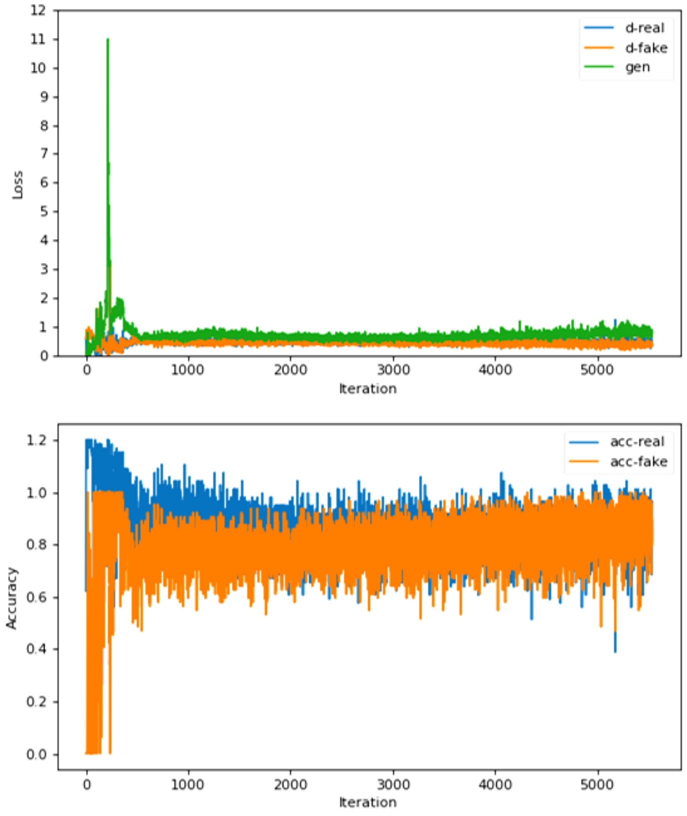


Figure 4.3: DCGAN Loss and Accuracy of Generator and Discriminator Network

4.1.1.1. Synthetic Image Quality Evaluation

Generative Adversarial Network (GAN) produces fake images as real as the original images. While these images appear similar to the naked eye there is certain evaluation required to confirm the realness of the generated images. To evaluate the augmented data, we first used Principle Component Analysis (PCA) to visualize the DCGAN augmented data in a 2D plane.

4.1.1.2. Principal Component Analysis

Principal Component Analysis (PCA) is unsupervised data visualization and dimensionality reduction technique for the data in high dimensional space. It is hard to get insight from the data that resides in a higher dimension and also is computationally very intensive. The aim of this technique is to decrease data dimensionality which is highly correlating by applying transformation on original vectors to obtain a new set called principal component. It tends to preserve the global structure in data by mapping the cluster as a whole due to which local structures in data might get lost. While PCA enables the representation of data by reducing dimensions, it retains eigen vectors and eigen components which are then projected onto eigen space to get the visualization. Application of PCA includes feature extractions, noise filtering, gene data analysis, and stock market predictions.

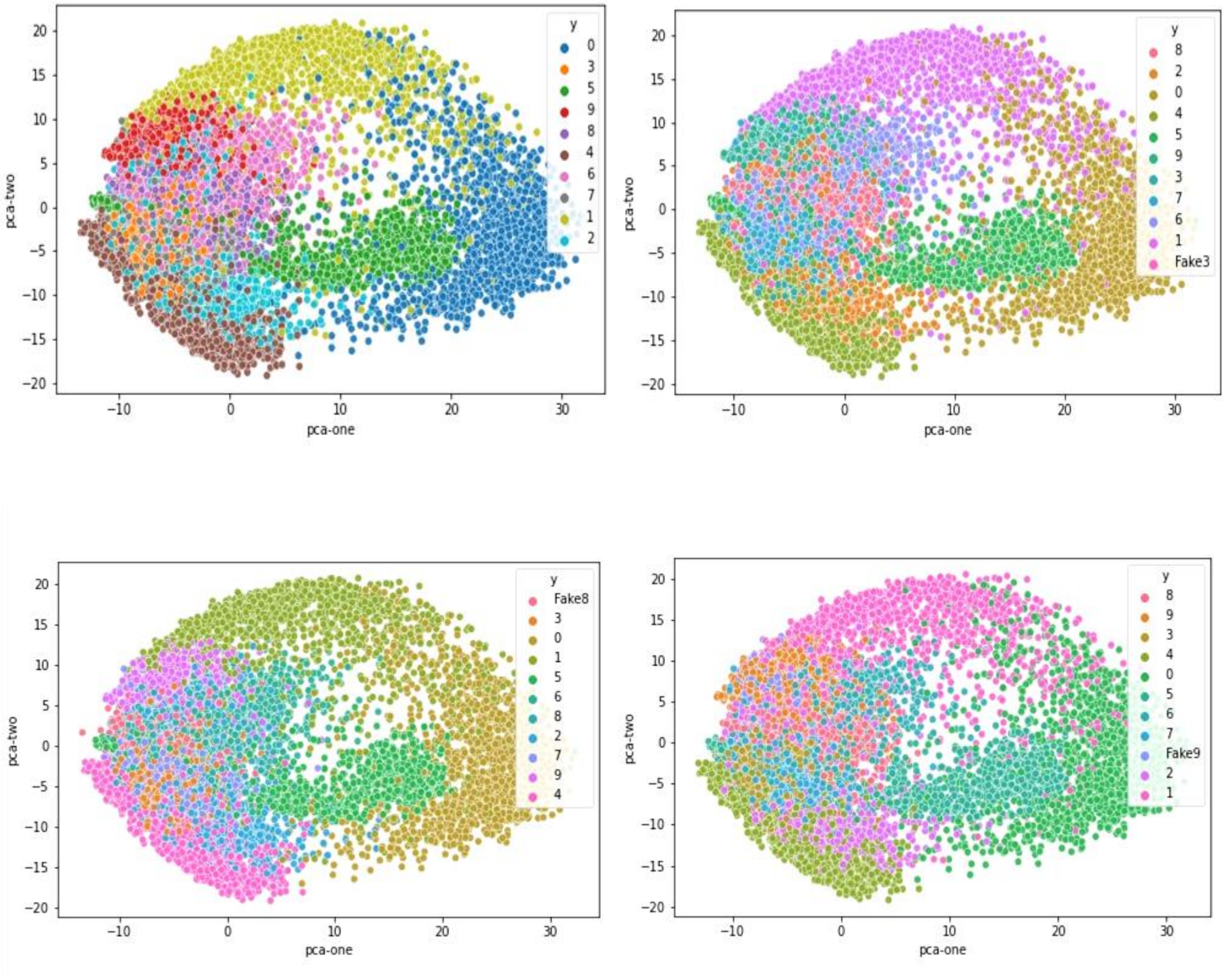


Figure 4.4: PCA (a) PCA of Original Dataset (b) PCA of DCGAN generated images of numeral 3. (b) PCA of DCGAN generated images of numeral 8. (d) PCA of DCGAN generated images of numeral 9.

The result of PCA, when applied to DCGAN, augmented data can be seen from Figure 4.3. We can observe that the distinction between numerals is not clear. One reason being the Linear algorithm so does not learn the complex polynomial features.

4.1.1.3. t-Stochastic Neighbor Embedding (t-SNE)

t-Stochastic Neighborhood Embedding (t-SNE) [68] is also a data visualization and an unsupervised dimensionality reduction technique but unlike PCA, t-SNE is the non-linear algorithm that used complex mathematics but the idea behind it is simple. It transforms the point from a higher dimension to a lower dimension while preserving the neighborhood of the point. It preserves the local structure between two distributions in data by reducing the Kullback–Leibler divergence (KL divergence) corresponding to the location of a point on the map. t-SNE finds application in music analysis, computer security research, cancer research, biomedical signal processing, and bioinformatics.

t-SNE is also used as a tool for evaluating the GAN augmented images. In many research works, it is the first used evaluation metric used for the evaluation of artificially generated images. In this thesis, we used the t-distributed stochastic neighbor embedding (t-SNE) approach to evaluate the realness of the fake images generated by DCGAN.

From Figure 4.4(b), (c), and (d) we can observe that the results of DCGAN in terms of distribution of data points are very close to the real images. When the samples generated using DCGAN for digits 3, 8, and 9 are visualized on a 2D plot, it is very evident that real and fake images for this digit are clustered together and it is very hard to distinguish between original and DCGAN augmented images due to their high correlation.

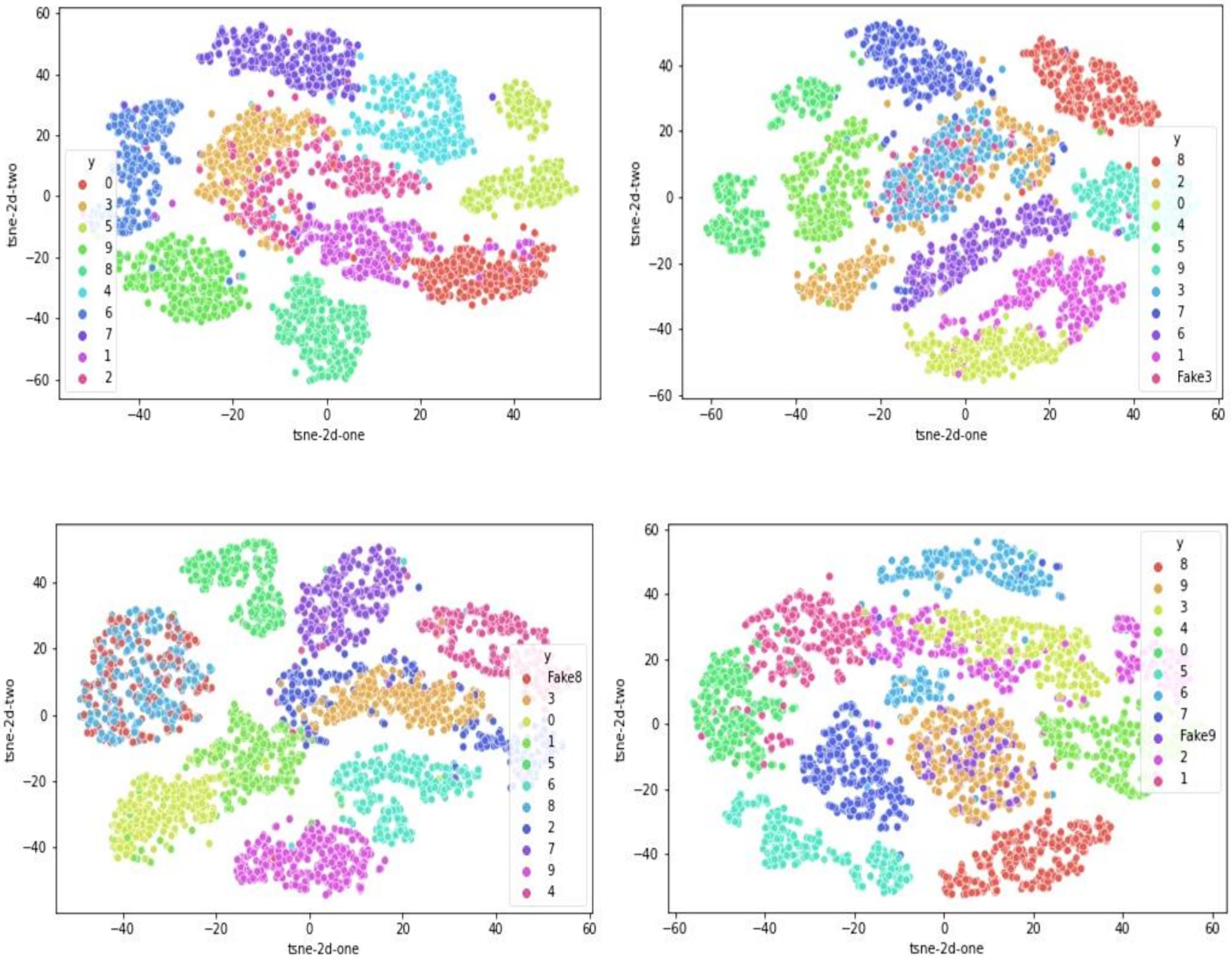


Figure 4.5: t-SNE (a) t-SNE of Original Dataset (b) t-SNE of DCGAN generated images of numeral 3. (b) t-SNE of DCGAN generated images of numeral 8. (d) t-SNE of DCGAN generated images of numeral 9.

4.2. Convolution Neural Network with Attention Mechanism

4.2.1. Experimentation and Results

We rigorously evaluated three architectures namely vanilla ResNet18, ResNet18 incorporating squeeze and excitation block, and convolution block attention module in ResNet18. All these architecture were experimented with and without DCGAN augmented images.

The first set of experiments was conducted without DCGAN artificially generated images. The dataset was split into three independent and identically distributed sets: Training, Validation, and

Test. Training data consist of 80% of the dataset comprising 7,840 images. Validation data comprised 10% of training data comprising 784 images while test data comprise 10% of the dataset, resulting in 1960 images. The three architectures were trained in training data and fine-tuned on validation data.

In the second set of experiments, we generated 30% new images which increased the total size of the dataset by 13,600. The training, validation, and test ratio used were the same as before. This resulted in 10,800 train images, 612 validation images, and 1360 test images.

We experimented using different batch sizes of 32, 64, and 128. While we used Adam optimizer because it sets regret bound on convergence rate [70] Adaptive learning is employed while training these architectures. The initial value of the learning rate is set to 1e-1 for the first 80 epochs. It then reduces to 1e-2 after 120 epochs and after 160 epochs further reduces to 1e-3. Finally for epochs greater than 180 learning rate of 0.5e-3 was used. The accuracy and loss graphs for six architectures are shown in Figure 4.5 and Figure 4.6. We can observe from the graphs that each experiment conducted was not subjected to overfitting and underfitting during training. The hyperparameters selected ensured the ideal fit which was only possible after extensive and repeated experiments.

The results are summarized in Table 4.1. We can clearly conclude that though none of the experiments performed below average on the test set the addition of DCGAN images increased the performance in the second set of experiments. In both sets of experiments, CBAM performed better than the vanilla version of ResNEet18 and SE version of ResNet18. And out of all six experiments ResNet18 with CBAM module with artificially generated images using DCGAN performed the best.

Table 4.1: Results of six Architectures

	ResNet18	**ResNet18 + SE**	**ResNet +CBAM**
Without DCGAN images	96.2%	97.5%	98.8%
With DCGAN images	97.2%	99.1%	100%

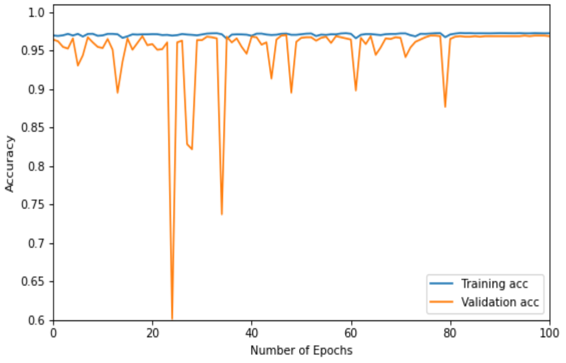
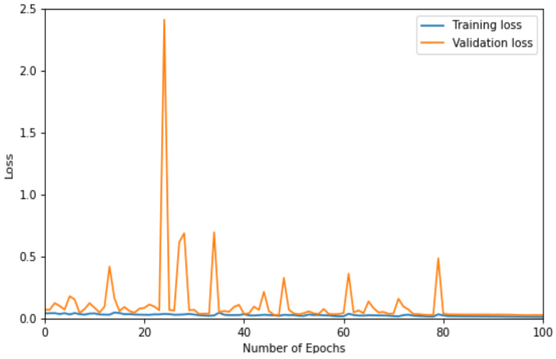
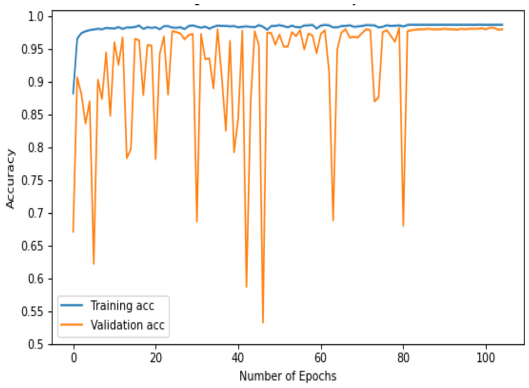
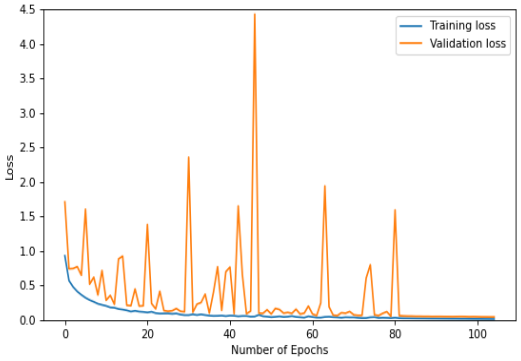
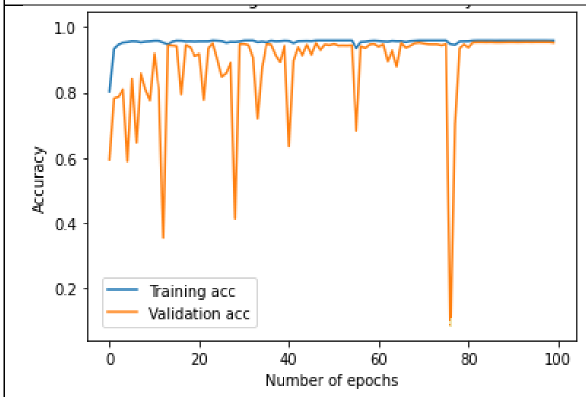
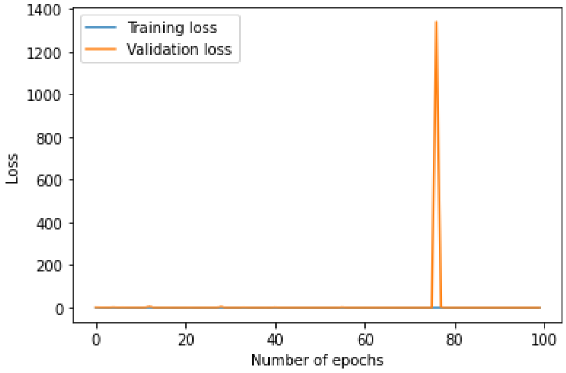
Without DCGAN images	
Accuracy	**Loss**
ResNet18	
	
ResNet18 + SE	
	
ResNet18 + CBAM	
	

Figure 4.6: Without DCGAN augmented imagesLoss and Accuracy graphs for (a) ResNet18 (b) ResNet18 +SE (c) ResNet18 +CBAM.

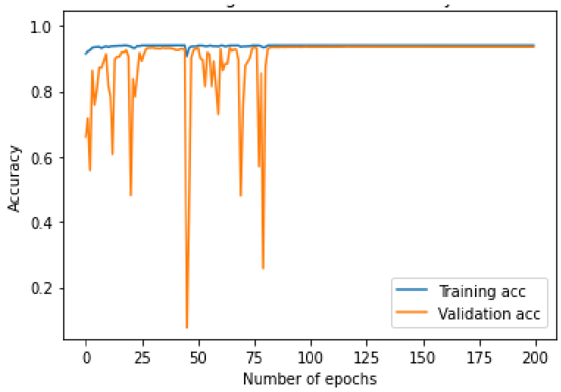
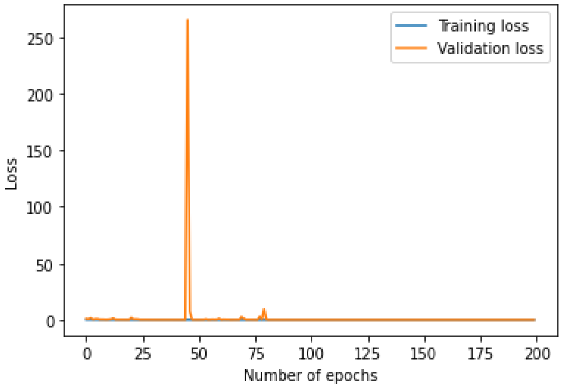
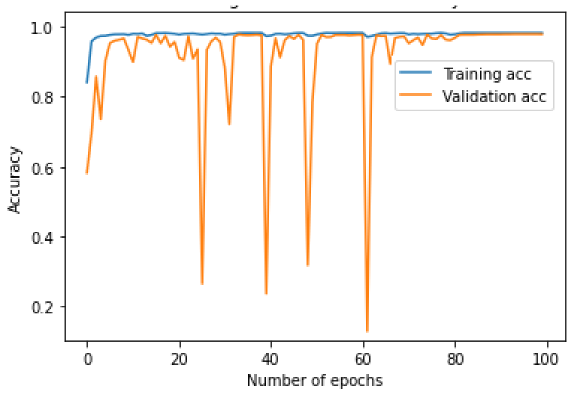
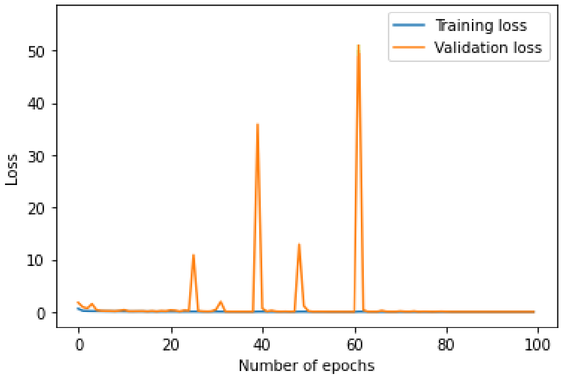
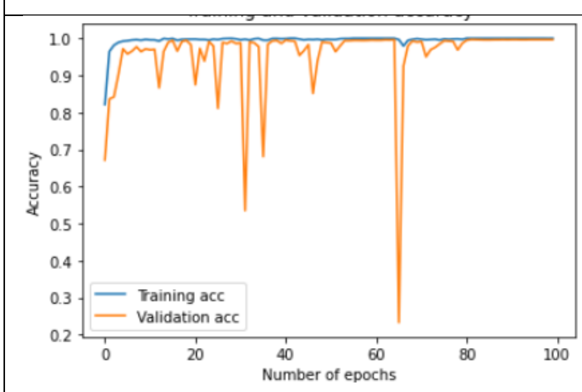
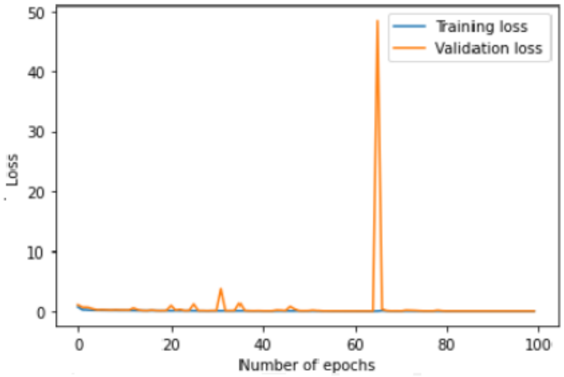
<table>
<tr><th colspan="2">With DCGAN images</th></tr>
<tr><th>Accuracy</th><th>Loss</th></tr>
<tr><td colspan="2"><b>ResNet18</b></td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2"><b>ResNet18 + SE</b></td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2"><b>ResNet18 + CBAM</b></td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Figure 4.7: Without DCGAN augmented images Loss and Accuracy graphs for (a) ResNet18 (b) ResNet18 +SE (c) ResNet18 +CBAM.

4.2.2. Evaluation

To evaluate the results we computed a confusion matrix. A confusion matrix is used to demonstrate the performance of the model used for classification. It used a test set for which we already know the true value. Since the test set is not seen by the model performance on the test set provides the performance of the model when deployed in the real world.

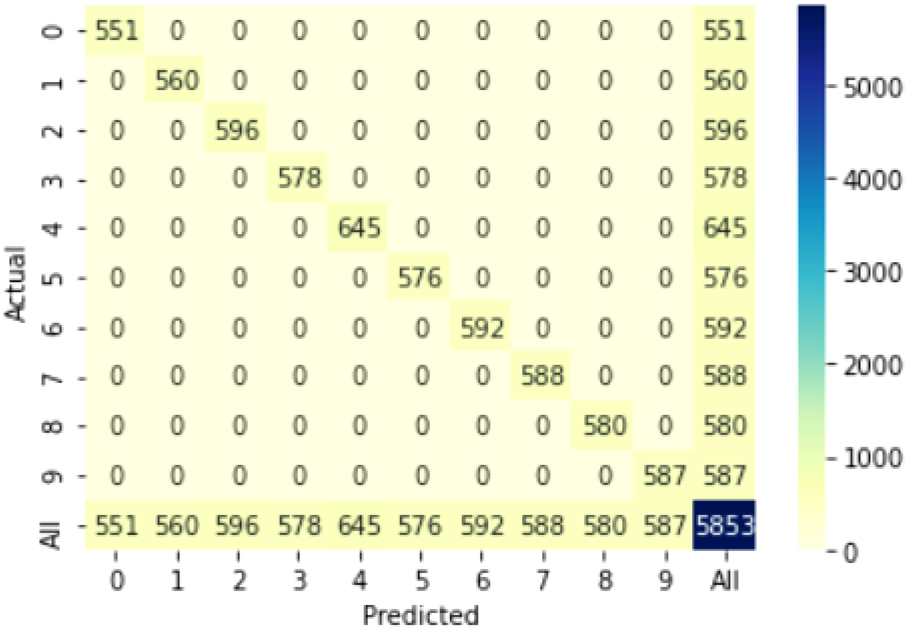


Figure 4.8: Confusion Matrix for CBAM Model with DCGAN augmented images

Figure 4.6 shows the performance of the ResNet18+CBAM model on the test set. We can see that out of 1360 test images only 4 images are wrongly classified. While others are correctly classified.

4.2.2.1. Evaluation using different Test Sets

Since we achieved the highest accuracy on the model trained using the attention mechanism on the dataset that now includes the DCGAN augmented images, we created different test sets obtained from various sources. We created four test sets from sources given below in Figure 4.9

- **Pakistan Currency Notes**

We took pictures of Pakistani Currency notes of Rs. 5000, 1000, 50, 20, 10. Then using connected component labeling we cropped the images of numerals 0, 1, 2, and 5 shown in Figure

4.9. We created the set consisting of 42 images. . Distribution of the currency test set is given in Table 4.2. We can see many images are of 0 because 0 is very abundant in currency notes.

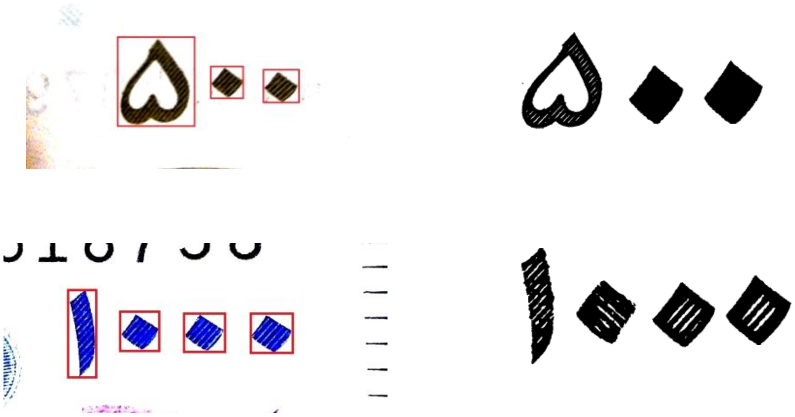


Figure 4.9: Pakistani Currency notes cropped using Connected Component Analysis

Table 4.2 Distribution of Pakistani Currency Notes

Numeral	**0**	**1**	**2**	**5**
Quantity	27	5	5	5

Once a test set was formed we then provided this set to trained model and observed the result shown in Figure 4.10.

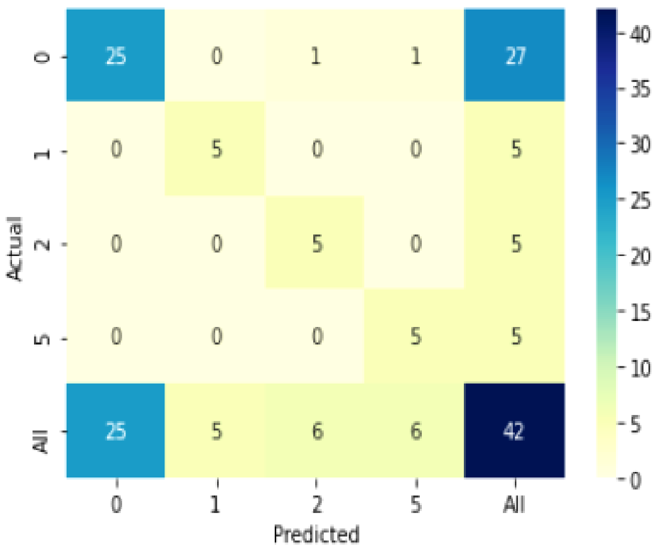


Figure 4.10: Confusion matrix of results obtained from Pakistani Currency Notes

On creating the confusion matrix we concluded that our model successfully classified the images to their respective class. Out of 42 images 40 are correctly classified i.e only two images of 0 are wrongly predicted as 2 and 5.Though there is the extreme variation found in the digits, for instance, the stroke in these images is bolder than the one found in our training set. Also, the digits of this currency are textured. Such texture is not found in any of the training images. Our model was able to predict even the textured numerals. This confirms the robustness of our model.

- **Handwritten Urdu Numerals**

In this test set, we wrote a handwritten Urdu numeral using a black marker on a white page. These images were written on similar lines as of the original dataset but were not a part of the training set hence they follow the different distribution. Hence they contain properties and characteristics that our model has not seen before. These numerals had different sizes and a variation in orientation as shown in Figure 4.11

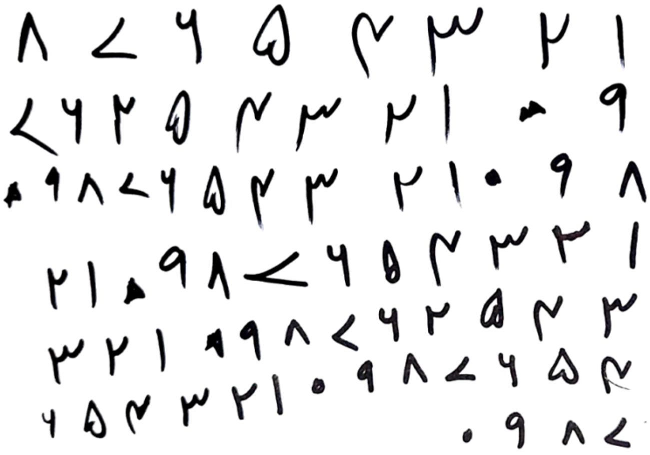


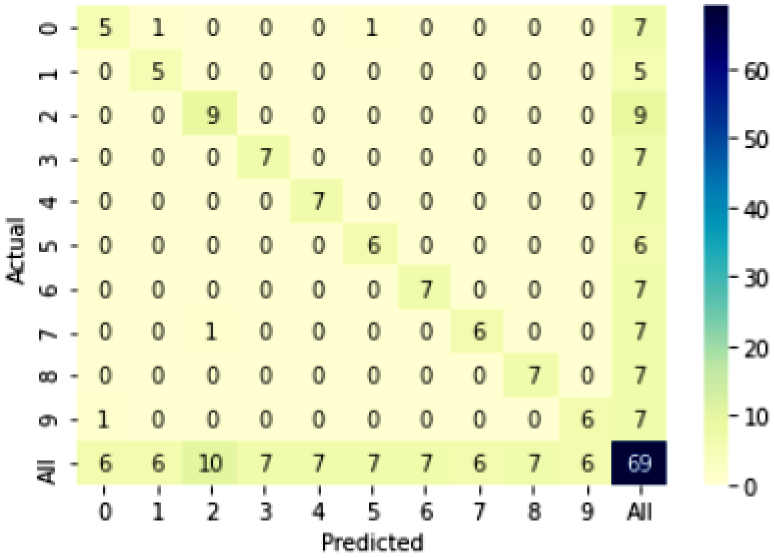
Figure 4.11: Handwritten Urdu Numeral Test Set

These images were cropped using the connected component labeling technique and were provided to our dataset. This dataset consisted of 69 images. Distribution is given in Table 4.3 below.

Table 4.3: Distribution of Handwritten Numeral Test Set

Numeral	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Quantity	7	5	9	7	7	6	7	7	7	7

The results of the trained model on this versatile dataset validated the performance of our model shown in Figure 4.12.



Actual \ Predicted	0	1	2	3	4	5	6	7	8	9	All
0	5	1	0	0	0	1	0	0	0	0	7
1	0	5	0	0	0	0	0	0	0	0	5
2	0	0	9	0	0	0	0	0	0	0	9
3	0	0	0	7	0	0	0	0	0	0	7
4	0	0	0	0	7	0	0	0	0	0	7
5	0	0	0	0	0	6	0	0	0	0	6
6	0	0	0	0	0	0	7	0	0	0	7
7	0	0	1	0	0	0	0	6	0	0	7
8	0	0	0	0	0	0	0	0	7	0	7
9	1	0	0	0	0	0	0	0	0	6	7
All	6	6	10	7	7	7	7	6	7	6	69

Figure 4.12: Confusion matrix of results obtained from Handwritten numerals

Out of 69 images of handwritten Urdu numerals only 4 are misclassified and 65 are correctly classified. Hence the model does not best not only on test data split from original data but also on the data that is obtained in similar but unrelated ways.

- **Numerals Written in thin strokes**

To further validate the performance of our training model we used another diversified test set that was written by hand using a thin nip marker. These numerals shown in Figure were entirely different from one found in our data

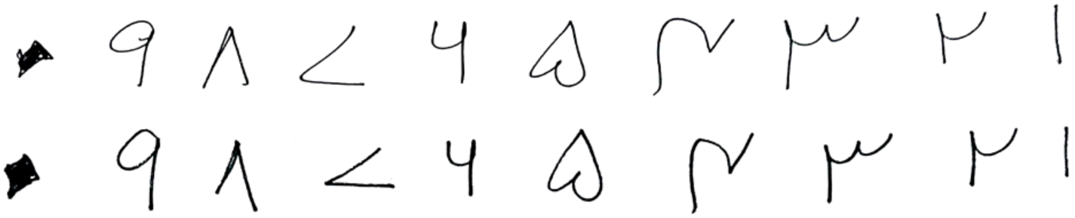


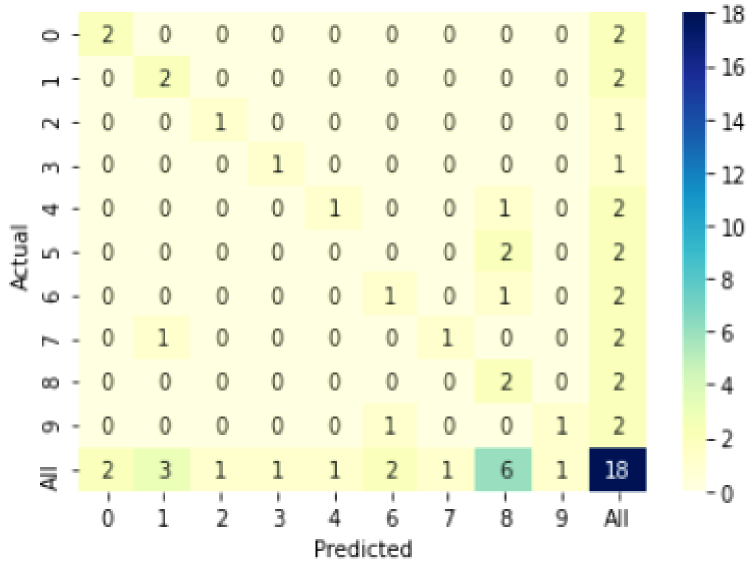
Figure 4.13: Thin stroked Numerals

The dataset contained 18 images. The distribution of numeral images is given in Table 4.4.

Table 4.4: Distribution of thin stroked Dataset

Numeral	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Quantity	2	2	1	1	2	2	2	2	2	2

We observed that only three of the images were misclassified even though the model did not see such data during training and these numerals are very different from the numerals of the original dataset. The confusion matrix of this dataset is elaborated in Figure 4.14.



Actual \ Predicted	0	1	2	3	4	6	7	8	9	All
0	2	0	0	0	0	0	0	0	0	2
1	0	2	0	0	0	0	0	0	0	2
2	0	0	1	0	0	0	0	0	0	1
3	0	0	0	1	0	0	0	0	0	1
4	0	0	0	0	1	0	0	1	0	2
5	0	0	0	0	0	0	0	2	0	2
6	0	0	0	0	0	1	0	1	0	2
7	0	1	0	0	0	0	1	0	0	2
8	0	0	0	0	0	0	0	2	0	2
9	0	0	0	0	0	1	0	0	1	2
All	2	3	1	1	1	2	1	6	1	18

Figure 4.14: Confusion matrix for thin stroked Numerals

- **Numerals obtained via touch**

To create this dataset we downloaded an application from Google Play Store called “Drawing Pad”. We wanted to see the performance of our model on the numerals that are not merely written using markers and papers on a sheet of paper. Our aim with this test set is to see the performance of the model when this is deployed in the application that uses a medium of touch to draw things. This dataset contained 21 images. The distribution is shown in Table 4.5

Table 4.5: Distribution of Gadgets Test Set

Numeral	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Quantity	2	2	1	1	3	2	2	1	2	3

We observed that out of 21 images only one image is misclassified. Hence this model is robust to touch as well which makes it very reliable when deployed in gadgets that use touch medium to generate numeral images. The result in form of a confusion matrix is shown in Figure 4.15.

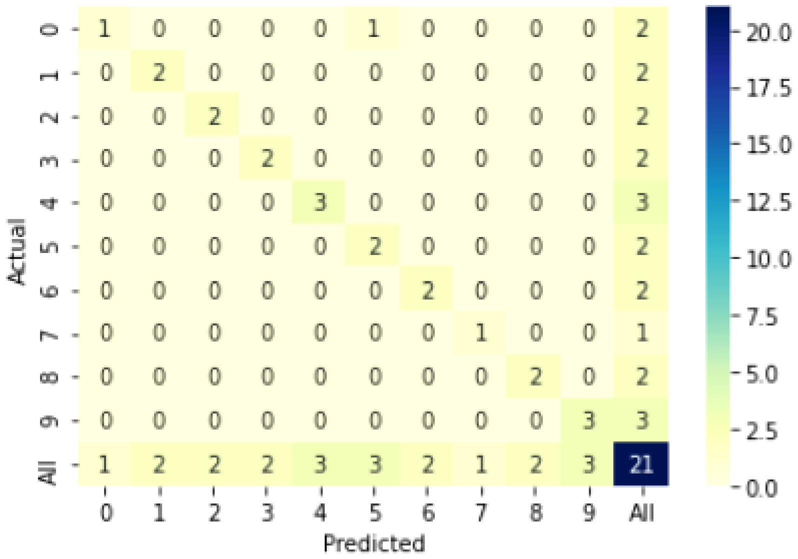


Figure 4.15: Confusion Matrix for gadget test set

After repeated experiments using various versatile test set that not only bring new challenges for the model but also map the non-linearity of the real-world, we prove that this model is robust and produce a state of the art result on numerals written in various style and platforms.

4.2.2.2. Evaluation on numerals of different languages

After experimenting on various test sets of Urdu language, we next tested the trained numeral classifier on different languages namely, Persian, Arabic, and English.

- **Persian**

To further validate the robustness of our model, we have experimented with the trained Urdu numeral classifier on Persian numerals. In comparison to the Urdu language, 8 out of 10 Persian numerals are very similar i.e., 0, 1, 2, 3, 5, 6, 8, and 9. Digits 4 and 7 are different in both languages. We have collected a dataset of 29 Persian numerals. The distribution of the dataset is shown in Table 4.6.

Table 4.6: Distribution of Persian numeral Test Set

Numeral	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Quantity	3	2	3	3	3	3	3	3	3	3

These numerals were written on paper with a marker and then cropped using connected component labeling. We observed that out of 29 images 23 images were correctly classified. As expected, all images digits 0, 1, 2, 3, 5, 6, 8, and 9 are correctly recognized by Urdu classifier. Hence we can conclude that the Urdu classifier can be used to make predictions on Persian digits however the result in the case of Persian numerals is not as reliable as in the case of Urdu. The result in form of a confusion matrix is shown in Figure 4.16.

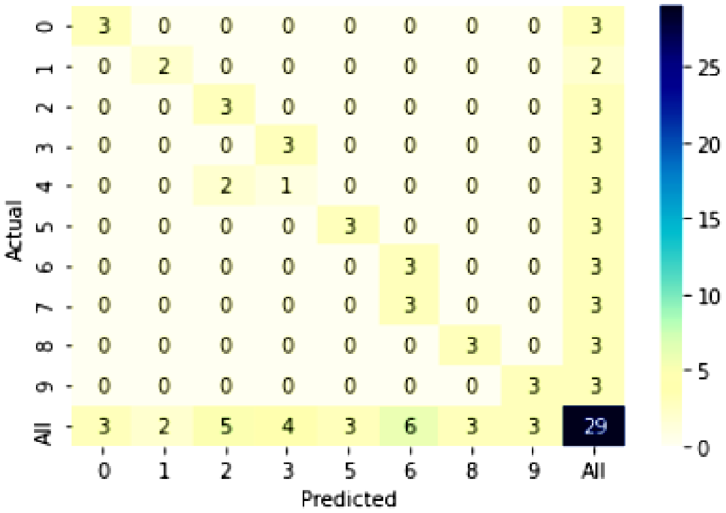


Figure 4.16: Confusion Matrix for Persian numeral Test Set

- **Arabic**

We also applied the trained Urdu numeral classifier to the Arabic language numeral. Out of 10 numerals, 5 in both languages are the same i.e., 0, 1, 3, 8, and 9. We wrote numerals on paper with a marker which were then cropped using connected component labeling. The distribution of the test set is given in Table 4.17.

Table 4.7: Distribution of Arabic numeral Test Set

Numeral	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Quantity	4	4	4	3	3	3	3	3	3	3

We observed that the digits that were similar in both languages, the classifier was able to correctly recognize them. However, the rest of the images was incorrectly recognized. Out of 33 total images, 18 images were correctly assigned labels. We conclude that this classifier cannot be used on Arabic language numeral satisfactory. The result in form of a confusion matrix is shown in Figure 4.17.

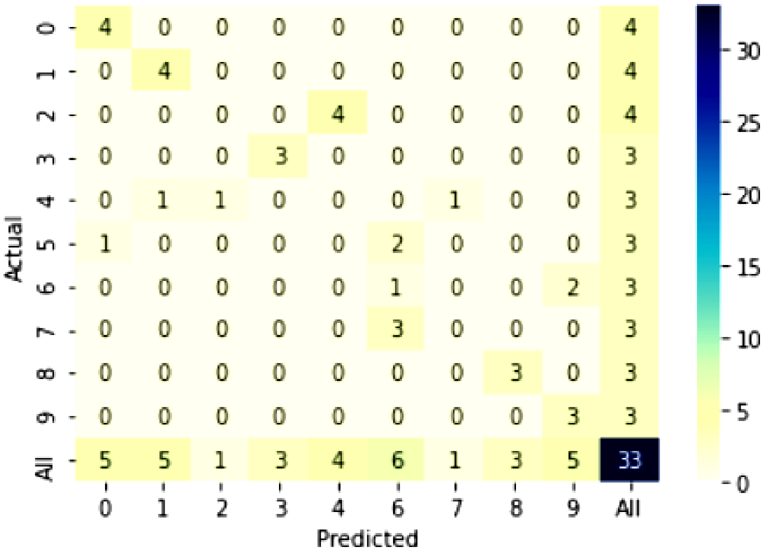


Figure 4.17: Confusion Matrix for Arabic numeral Test Set

- **English**

We then applied a trained Urdu numeral classifier to make a prediction on English numerals. In comparison to Urdu numerals, out of 10, only two numerals are similar to English numerals i.e., 1 and 9. The rest of the numerals is very different in both languages. To create an English numeral test set, 49 images were written on paper with a marker which was then cropped using connected component labeling. The distribution is given in Table 4.8.

Table 4.8: Distribution of English numeral Test Set

Numeral	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Quantity	3	2	3	3	3	3	3	3	3	3

Out of 49 images only 10 images were correctly classified. We observed that the numerals that were similar to Urdu numerals in the English language were correctly classified that is the images of 2 out of 10 digits were correctly classified. Hence, the result of the Urdu numeral classifier on English numeral is unsatisfactory and cannot be used for classifying the digits of the English language. The result in form of a confusion matrix is shown in Figure 4.18.

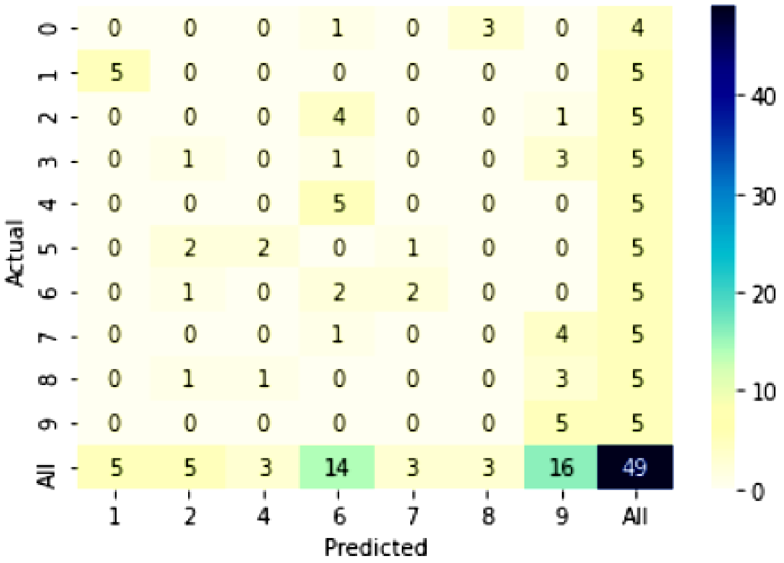


Figure 4.18: Confusion Matrix for English numeral Test set

We have provided a comparison of accuracies when Persian, Arabic, and English numerals were tested on trained Urdu numeral classifiers in Table 4.9. We observe that the more similar the numerals are with the Urdu language the higher is the accuracy.

Table 4.9: Accuracies comparison of Persian, Arabic and English numeral

	Persian	**Arabic**	**English**
Accuracy	79.3%	54.5%	18.4%

5 Conclusion and Future Work

The Urdu language is an amalgam of various languages that is very popular in South East Asia as well as in the Middle East. It is also the national language of Pakistan. Hence Urdu numeral finds many applications here. Even today many forms of field such as date of birth use Urdu numerals. As per a survey conducted in 2018, 63.1% population is urban hence writing and speaking English is not feasible for developing countries like Pakistan. Therefore there is a severe need for research in our national language Urdu. Urdu numerals find application in Pakistani currency notes, Pakistani postage stamps, numbering and indexing the chapters in Holy Quran.

One reason behind the excel of research in a particular domain is the data. Easy access to data can produce state of art results in any domain. Major scripts of famous languages have been researched upon due to large public datasets. However, there is no public dataset available for Urdu numerals. In this thesis, we present a novel Urdu numeral dataset that is acquired following the latest practices. The data was acquired from people of various age groups and social circles so that more variety can be added to the dataset. Since every person has a unique style of writing this dataset also includes the hard cases such as crumbling the paper and adding background noises while writing the Urdu digits. Since the need for the data is always there. The deeper the network more data is required to train it. In order to generate data for this particular domain, we tried Deep Convolutional Generative Adversarial Network (DCGAN). However, the result generated were not as real as we wanted them to be hence we augmented the numeral images for which DCGAN found it hard to train. After provided the augmented dataset to DCGAN we noticed that numerals were not only generated but were as real as the original dataset. To further validate the result of artificially generated DCGAN images we applied Principal Component Analysis (PCA) t-distributed Stochastic Neighbour Embedding (t-SNE). They provide the visualization of the digits in the 2D plane. We observed that fake and real images had a high correlation. They not only belong to the same clusters but it is hard to tell them apart.

Once we have generated the artificial images we next develop the state of art classifier. We perform three sets of experiments with and without DCGAN generated fake images. We choose

ResNet 18 model, a combination of Squeeze and Excitation Block (SE) with ResNet18 and a combination of Convolution Block of Attention Module with ResNet 18. After repeated experiments, we see that adding DCGAN images positively increased the performance. We initially added 30% of the artificial data and using CBAM with ResNet18 performed state-of-the-art results. We achieved an accuracy of 100% on test data. To further validate the result we created a test set obtained from four different sources. We first tested the model with 27 images of Pakistani currency notes that contained classes 0, 1, 2, and 5.We observed that only 2 images were misclassified though this dataset contained textured images. We then tested this model on a handwritten Urdu numeral that had different sizes and slant orientations. We observed that again only 4 images out of 69 are misclassified validating the robustness of our model. Next, we tested this model using another test set of 18 images that contained numerals written with thin strokes. While they provided new challenges to our model as these images were very different from the one our model had been trained on. Out of 18 only 3 images were misclassified confirming the robustness of the model trained on the Urdu numeral dataset. We next tested the model on Urdu numeral gathered using touch medium instead of paper pen and marker owing to the fact since we wanted to see the performance of the model when deployed on gadgets application. Out of 21, only one image was misclassified. Hence we confirm after repeated experiments on various datasets obtained from versatile sources that our Urdu numeral classifier produces state-of-the-art results. We also experimented with the robustness of the Urdu numeral classifier on the numeral of various languages i.e., Persian, Arabic, and English. We observed that Persian and Urdu numerals have 8 numerals common and hence highest accuracy of 79.3% has been achieved on Persian numerals. Since the number of similar numerals in Arabic and Urdu is lesser i.e, 5, the accuracy achieved is less in this case that is 54.5%. However, the least accuracy of 18.4% is achieved for English numerals because only 2 numerals are similar in the English and Urdu language. Hence we conclude this model cannot be used for Arabic and English language. However, the use of the Urdu numeral classifier for Persian numerals is debatable.

5.1 Future Work

This model can revive the Urdu language that has been declining with time. Since Urdu is the national language of Pakistan this research can give birth to Urdu recognizers that when employed can solve application that uses the paper medium as well as touch medium.

Such a trained model can be deployed in the real world to solve many problems. In this time of global pandemic when online classes have become new normal, we can make use of this model for automatic dictation. Such application can give rise to a paperless environment where beginners can learn Urdu numerals and the model automatically classify whether the written word resembles any of the numerals. Since the performance of deep learning algorithms in real-world applications is of utmost importance, so we plan on testing it on other applications as well such as recognizing the Surah number of The Holy Quran, numbers on Pakistani currency notes, and on Pakistani postage stamps.

Since here we have only used DCGAN while there is another variety of GANs available. In the future, we aim to experiment with other GANS. We have only used 30% of new images. There is a need for further experimentation to see how much more images can be generated before the performance starts degrading. Furthermore, as new models are being developed we can experiment with other models which are less deep to achieve the same performance but less number of parameters.

References:

[1] Simons, G.F.; Fennig, C.D. Ethnologue: Languages of Asia; Sil International: Dallas, TX, USA, 2017.

[2] Husnain, M.; Missen, M.M.S.; Mumtaz, S.; Jhanidr, M.Z.; Coustaty, M.; Muzzamil Luqman, M.; Ogier, J.M.; Choi, G.S. Recognition of Urdu Handwritten Characters Using Convolutional Neural Network, Applied Sciences, 2019, doi: 10.3390 9132758.

[3] Shah, Aman. (2016). Teaching of Urdu: Problems and Prospects.

[4] Cecotti, H. (2016). Active graph based semi-supervised learning using image matching: Application to handwritten digit recognition. Pattern Recognition Letters,73, 76-82. doi:10.1016/j.patrec.2016.01.016

[5] Elleuch, M., Maalej, R., & Kherallah, M. (2016). A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition. Procedia Computer Science,80, 1712-1723. doi:10.1016/j.procs.2016.05.512

[6] Niu, X., & Suen, C. Y. (2012). A novel hybrid CNN–SVM classifier for recognizing handwritten digits. Pattern Recognition,45(4), 1318-1325. doi:10.1016/j.patcog.2011.09.021

[7] https://en.wikipedia.org/wiki/2017_Census_of_Pakistan

[8] Pramanik, R., & Bag, S. (2018). Shape decomposition-based handwritten compound character recognition for Bangla OCR. Journal of Visual Communication and Image Representation,50, 123-134. doi:10.1016/j.jvcir.2017.11.016

[9] Sarkhel, R., Das, N., Saha, A. K., & Nasipuri, M. (2016). A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition. Pattern Recognition,58, 172-189. doi:10.1016/j.patcog.2016.04.010

[10] Biswas, M., Islam, R., Shom, G. K., Shopon, M., Mohammed, N., Momen, S., & Abedin, A. (2017). BanglaLekha-Isolated: A multi-purpose comprehensive dataset of Handwritten Bangla Isolated characters. Data in Brief,12, 103-107. doi:10.1016/j.dib.2017.03.035

[11] Nagajyothi, D. & Siddaiah, P.. (2018). Speech Recognition Using Convolutional Neural Networks. International Journal of Engineering and Technology(UAE). 7. 133-137. 10.14419/ijet.v7i4.6.20449.

[12] Sharma, Manik & Anuradha, J. & Manne, H & Kashyap, G. (2017). Facial detection using deep learning. IOP Conference Series: Materials Science and Engineering. 263. 042092. 10.1088/1757-899X/263/4/042092.

[13] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS), volume 2, pages 3104–3112. MIT Press. (page 1, 5) [71]

[14] Anwar, Syed & Majid, Muhammad & Qayyum, Adnan & Awais, Muhammad & Alnowami, Majdi & Khan, Khurram. (2018). Medical Image Analysis using Convolutional Neural Networks: A Review. Journal of Medical Systems. 42. 226. 10.1007/s10916-018-1088-1.

[15] Grigorescu, Sorin & Trasnea, Bogdan & Cocias, Tiberiu & Macesanu, Gigel. (2019). A survey of deep learning techniques for autonomous driving. Journal of Field Robotics. 37. 10.1002/rob.21918.

[16] Jiuxiang Gua, Zhenhua Wangb, Jason Kuenb, ”Recent Advances in Convolutional Neural Networks”, 2017

[17] Shanqing Gu, Manisha Pednekar, Robert Slater, ”Improve Image Classification Using Data Augmentation and Neural Networks”, 2019

[18] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of Wasserstein GANs, Advances in Neural Information Processing Systems 30 (NIPS), pages 5769–5779. Curran Associates, Inc.

[19] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. Proceedings of the Sixth International Conference on Learning Representations (ICLR).

[20] Bhattarai, Binod & Baek, Seungryul & Bodur, Rumeysa & Kim, Tae-Kyun. (2019). Sampling Strategies for GAN Synthetic Data. 10.1109/ICASSP40776.2020.9054677.

[21] J. Liu, Q. Li, P. Zhang, G. Zhang and M. Liu, "Unpaired Domain Transfer for Data Augment in Face Recognition," in IEEE Access, vol. 8, pp. 39349-39360, 2020, doi: 10.1109/ACCESS.2020.2976207.

[22] Woo, Sanghyun & Park, JongChan & Lee, Joon-Young & Kweon, Inso. (2018). CBAM: Convolutional Block Attention Module.

[23] Singh, R.; Mishra, R. K.; Bedi, S.; Kumar, S.; Shukla, A. K. ALiterature Review on Handwritten Character Recognition based on Arti-ficial Neural Network.International Journal of Computer Sciences andEngineering,2018, 6(11), 753-758. doi: 10.26438/ijcse/v6i11.753758.

[24] I. Ahmad, X. Wang, R. Li and S. Rasheed, ”Offline Urdu Nastaleeqoptical character recognition based on stacked denoising autoencoder”,China Communications, vol. 14, no. 1, pp. 146-157, 2017

[25] N. Javed, S. Shabbir, I. Siddiqi and K. Khurshid, ”Classification of UrduLigatures Using Convolutional Neural Networks - A Novel Approach,”2017 International Conference on Frontiers of Information Technology(FIT), Islamabad, 2017, pp. 93-97, doi: 10.1109/FIT.2017.00024.

[26] G. S. Lehal, ”Ligature Segmentation for Urdu OCR,” 2013 12th Interna-tional Conference on Document Analysis and Recognition, Washington,DC, 2013, pp. 1130-1134, doi: 10.1109/ICDAR.2013.229

[27] Kumar, G.; Bhatia, P. K. Analytical Review of Preprocessing Techniques for Offline Handwritten Character368Recognition.2nd International Conference on Emerging Trends in Engineering and Management ICETEM,2013

[28] Akhtar, P. An Online and Offline Character Recognition Using Image Processing Methods-A Survey Mr. 2016

[29] Liu, C.; Yin, F.; Wang, D.; Wang, Q. Online and offline handwritten Chinese character recognition:Benchmarking on new databases. Pattern Recognition2012, 46(1), 155-162. doi: 10.1016/j.patcog.2012.06.021

[30] Khan, S. A Mechanism for Offline Character Recognition.International Journal For Research In Applied Science And Engineering Technology2019, 7(4), 1086-1090. doi: 10.22214/ijraset.2019.4194

[31] Kour, H.; Gondhi, N. K. Machine Learning approaches for Nastaliq style Urdu handwritten recognition: A survey6th International Conference on Advanced Computing and Communication Systems (ICACCS)2020, pp. 50-54. doi:10.1109/ICACCS48705.2020.9074294.

[32] Malik, S.; Khan, S. A. Urdu online handwriting recognition. Proceedings of the IEEE Symposium on Emerging Technologies, Islamabad, Pakistan, 2005. doi: 10.1109/ICET.2005.1558849.

[33] 16.Javed, L.; Shafi, M.; Khattak, M.I,; Ullah, N. Hand-written Urdu Numerals Recognition Using Kohonen Self Organizing Maps 2019.

[34] Ahmed, S. B.; Hameed, I. A.; Naz, S.; Razzak, M. I.; Yusof, R. Evaluation of Handwritten Urdu Text by Integration of MNIST Dataset Learning ExperienceIEEE Access 2019,vol. 7, pp. 153566-153578. doi:40210.1109/ACCESS.2019.2946313

[35] Gautam N.; Sharma R.S.; Hazrati G. Eastern Arabic Numerals: A Stand out from Other JargonsInternational Conference on Computational Intelligence and Communication Networks (CICN)2015, pp. 337-338. doi: 10.1109/CICN.2015.73.

[36] Abdelazeem, S. Comparing Arabic and Latin Handwritten Digits Recognition Problems.International Journal439of Computer and Information Engineering2009, pp. 1583 - 1587. doi: doi.org/10.5281/zenodo.1060383.

[37] Abdelazeem, S.; El-Sherif, E. The Arabic Handwritten Digits Databases ADBase & MADBase. Available441online: http://datacenter.aucegypt.edu/shazeem/ (accessed on: 14 May 2020)

[38] Das, N.; Sarkar, R.; Basu, S.; Kundu, M.; Nasipuri, M.; Basu, D.K. A genetic algorithm based region sampling454for selection of local features in handwritten digit recognition application.Applied Soft ComputingVolume45512, Issue 5, 2012

[39] Baldominos, A.; Saez, Y.; Isasi, P. Evolutionary Convolutional Neural Networks: An Application to404Handwriting Recognition.Neurocomputing2018, 283, 38–52. doi: 10.1016/j.neucom.2017.12.049.

[40] Alharbi, Rawan & Almajnooni, Nujood & Albishry, Maani & Alotaibi, Arwa & Alsaadi, Ferial & Alsulami, Gadeer & Alharbi, Amaal & Alotaibi, Renad & Alharbi, Rawabi & Alharbi, Ashwaq & Alkhodidi, Tahani & Alafif, Tarik & Albassam, Ayman & Sabban, Sari. (2020). GEAD: Generating and Evaluating Handwritten Eastern Arabic Digits Using Generative Adversarial Networks. 10.13140/RG.2.2.24679.06561.

[41] Haque, Sadeka & Shahinoor, Shammi & Rabby, Akm Shahariar Azad & Abujar, Sheikh & Hossain, Sayed. (2019).

[42] OnkoGan: Bangla Handwritten Digit Generation with Deep Convolutional Generative Adversarial Networks. 10.1007/978-981-13-9187-3_10.

[43] Jha, G., & Cecotti, H. (2020). Data augmentation for handwritten digit recognition using generative adversarial networks. Multimedia Tools and Applications. doi:10.1007/s11042-020-08883-w

[44] Tran, N.T., Tran, V.H., Nguyen, N.B., Nguyen, T.K., & Cheung, N.M. (2021). On Data Augmentation for GAN Training. *IEEE Transactions on Image Processing, 30, 1882–1897.*

[45] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, “Image augmentations for gan training,” arXiv preprint arXiv:2006.02595, 2020.

[46] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. CoRR, abs/1409.3215, 2014.

[47] A. Graves. Generating sequences with recurrent neural networks. In Arxiv preprint arXiv:1308.0850, 2013.

[48] Z. Guan, K. G. Yager, D. Yu and H. Qin, ”Multi-Label Visual Feature Learning with Attentional Aggregation,” 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO,USA, 2020, pp. 2190-2198, doi: 10.1109/WACV45572.2020.9093311.

[49] Wang, Fei, et al. ”Residual attention network for image classification.”Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[50] J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, ”Squeeze-and-Excitation Networks,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 2011-2023, 1 Aug. 2020, doi:10.1109/TPAMI.2019.2913372

[51] Baker, P.; Hardie, A.; McEnery, T.; Cunningham, H.; Gaizauskas, R.J. EMILLE, A 67-Million Word Corpus of Indic Languages: Data Collection, Mark-up and Harmonisation. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC’02) LREC, Las Palmas, Canary Islands Spain, May 2002.

[52] Sabbour, N.; Shafait, F. A segmentation-free approach to Arabic and Urdu OCR. Proc. SPIE 8658, Document 347 Recognition and Retrieval XX, 86580N, 4 February 2013. doi: 10.1117/12.2003731. 348

[53] Center for Language Engineering Urdu Ligatures from Corpus Page. Available online: http://www.cle.org. 345 pk/software/ling_resources/UrduLigaturesfromCorpus.htm (accessed on 11 June 2020). 346

[54] Slimane, F.; Kanoun, S.; Hennebert, J.; Alimi, A.; Ingold, R. A study on font-family and font-size recognition 349 applied to Arabic word images at ultra-low resolution. Pattern Recognition Letters 2013. 34(2), 209-218. doi: 350 10.1016/j.patrec.2012.09.012. 351

[55] Ahmed, S.; Naz, S.; Swati, S.; Razzak, M. Handwritten Urdu character recognition using one-dimensional 352 BLSTM classifier. Neural Computing And Applications 2017. 31(4), 1143-1151. doi: 10.1007/s00521-017-3146-x.

[56] Kora, Sagar & Ravula, Sridhar. (2020). Evaluation of Deep Convolutional Generative Adversarial Networks for Data Augmentation of Chest X-ray Images. Future Internet. 13. 8. 10.3390/fi13010008.

[57] Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks.

[58] arXiv 2015, arXiv:1511.06434.

[59] Lu, K.L.; Chu, T.H. An Image-Based Fall Detection System for the Elderly. Appl. Sci. 2018, 8, 1995.

[60] Katircioglu, I.; Tekin, B.; Salzmann, M.; Lepetit, V.; Fua, P. Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism. Int. J. Comput. Vis. 2018, 126, 1326–1341.

[61] Liu, J.; Gu, C.K.; Wang, J.; Kim, H.J. Multi-scale multi-class conditional generative adversarial network for

[62] handwritten character generation. J. Supercomput. 2017, 12, 1–19.

[63] J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, "Squeeze-and-Excitation Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 2011-2023, 1 Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.

[64] Misra, D., 2021. Channel Attention \& Squeeze-and-Excitation Networks | Paperspace Blog. [online] Paperspace Blog. Available at: <https:\/\/blog.paperspace.com/channel-attention-squeeze-and-excitation-networks/> [Accessed 3 February 2021].

[65] N. Martinel, G. L. Foresti and C. Micheloni, "Deep Pyramidal Pooling With Attention for Person Re-Identification," in IEEE Transactions on Image Processing, vol. 29, pp. 7306-7316, 2020, doi: 10.1109/TIP.2020.3000904

[66] He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2015). Deep Residual Learning for Image Recognition. 7.

[67] Hu, Jie et al. “Squeeze-and-Excitation Networks.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2020): 2011-2023.

[68] Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research 9(Nov), 2579–2605 (2008)

[69] Zhu, Xinyue & Liu, Yifan & Li, Jiahong & Wan, Tao & Qin, Zengchang. (2018). Emotion Classification with Data Augmentation Using Generative Adversarial Networks. 10.1007/978-3-319-93040-428

[70] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.