UNIVERSITY OF CAPE COAST

COLLEGE OF AGRICULTURE AND NATURAL SCIENCES

SCHOOL OF PHYSICAL SCIENCES

TRANSFER LEARNING USING GOOGLENET CONVOLUTIONAL NEURAL NETWORKS FOR FACE RECOGNITION

BY

OSAFO SHIRLEY

PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF PHYSICS OF THE SCHOOL OF PHYSICAL SCIENCES, COLLEGE OF AGRICULTURE AND NATURAL SCIENCES, UNIVERSITY OF CAPE COAST, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR AWARD OF BACHELOR OF SCIENCE DEGREE IN ENGINEERING PHYSICS.

NOVEMBER 2022

# DECLARATION

**Candidate’s Declaration**

I hereby declare that this thesis is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

**Candidate Name: SHIRLEY OSAFO**

Signature ……………………… Date…………………..

**Supervisor’s Declaration**

I hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision of thesis laid down by the University of Cape Coast.

**Supervisor Name: PROF. MOSES JOJO EGHAN and DR. CHARLES L.Y. AMUAH**

Signature ……………………… Date………………………..

# ABSTRACT

The creation of a facial recognition system has attracted the attention of numerous researchers, in the discipline of artificial intelligence (AI). Facial recognition was created to serve a variety of reasons, such as biometric payment of bills, for security system and crime identification, access to restricted areas and so on.

In the context of facial recognition, this project compares the accuracy of photos that have been enhanced and those that have not. The difficulties in achieving high performance really stem from hardware constraints and a lack of training data sets. In order to increase the performance of the face-recognition system even for a lower number of photos, the Deep Transfer Learning approach employing GoogLeNet's pre-trained CNN was used in this study. The results showed the improvement in over-fitting and the accuracy in performance after using the data augmentation and GoogLeNet technique, the Progressive graph showed 90.91% .

# ACKNOWLEDGEMENT

I wish to express my ultimate gratitude to the Almighty God and also to my supervisor, Prof. Moses Jojo Eghan for his untiring supervision, motivation and support. Also, I say a very big thank you to all my lecturers at the Physics Department, University of Cape Coast.

Finally, I am grateful to my family for the love and care they showed me throughout this journey.

# DEDICATION

I dedicate this project report to the OSAFO family and friends.

# TABLE OF CONTENT

[DECLARATION ii](#_Toc122431445)

[ABSTRACT iii](#_Toc122431446)

[ACKNOWLEDGEMENT iv](#_Toc122431447)

[DEDICATION v](#_Toc122431448)

[TABLE OF CONTENT vi](#_Toc122431449)

[LIST OF FIGURES viii](#_Toc122431450)

[LIST OF ABREVIATIONS ix](#_Toc122431451)

[KEY WORDS x](#_Toc122431452)

[CHAPTER 1 1](#_Toc122431453)

[INTRODUCTION 1](#_Toc122431454)

[Background 1](#_Toc122431455)

[Objective 3](#_Toc122431456)

[Significants of study 4](#_Toc122431457)

[CHAPTER 2 5](#_Toc122431458)

[LITERATURE REVIEW 5](#_Toc122431459)

[Introduction 5](#_Toc122431460)

[Transfer Learning 5](#_Toc122431461)

[Artificial Neural Network 7](#_Toc122431462)

[Convolutional Neural Network 9](#_Toc122431463)

[Data Augmentation 9](#_Toc122431464)

[Related papers 11](#_Toc122431465)

[CHAPTER 3 17](#_Toc122431466)

[METHODOLOGY 17](#_Toc122431467)

[Introduction 17](#_Toc122431468)

[GoogleNet Architecture 17](#_Toc122431469)

[Transfer learning and training Process. 20](#_Toc122431470)

[1. Dataset 20](#_Toc122431471)

[2. Importing and Loading Data 20](#_Toc122431472)

[3. Load Pre trained Network 21](#_Toc122431473)

[4. Replace Final Layers or Add layer Branches 23](#_Toc122431474)

[5. Train Network and Augmentation of Data 25](#_Toc122431475)

[Convolutional Neural Network (CNN) Architecture 26](#_Toc122431476)

[CHAPTER 4 30](#_Toc122431477)

[RESULTS AND DISCUSSION 30](#_Toc122431478)

[Introduction 30](#_Toc122431479)

[OVERVIEW 30](#_Toc122431480)

[RESULTS 31](#_Toc122431481)

[DISCUSSION 32](#_Toc122431482)

[CHAPTER 5 35](#_Toc122431483)

[CONCLUSION AND RECOMMENDATION 35](#_Toc122431484)

[CONCLUSION 35](#_Toc122431485)

[RECOMMENDATION 35](#_Toc122431486)

[REFERENCES 36](#_Toc122431487)

# 

# LIST OF FIGURES

[Figure 1: Representation of full Google net architecture 18](#_Toc122430492)

[Figure 2: Detailed architecture and parameters 19](#_Toc122430493)

[Figure 3: Importing and loading data 21](#_Toc122430494)

[Figure 4: Loading pretrained network 22](#_Toc122430495)

[Figure 5: Adding layers and Extracting layer graph 24](#_Toc122430496)

[Figure 6: Training and Augmentation Section 25](#_Toc122430497)

[Figure 7: Training Progress Graph for Accuracy and Loss 31](#_Toc122430498)

[Figure 8: Results from webcam snapshot 31](#_Toc122430499)

[Figure 9: Results from Augmentation and Testing 32](#_Toc122430500)

# LIST OF ABREVIATIONS

Convolutional Neural Network – (CNN)

Artificial Neural Network – (ANN)

Artificial Intelligence -- (AI)

ImageNet Large-Scale Visual Recognition Challenge -- (ILSVRC)

Linear Discriminant Analysis -- (LDA)

Principal Component Analysis -- (PCA)

# KEY WORDS

Convolutional Neural Network CNN; Data Augmentation; Face Recognition; Transfer Learning;

# CHAPTER 1

# INTRODUCTION

This chapter reports on the background of study, problem statement, objective, and the significance of the project

## Background

Facial Recognition has long been in existence, Beginning as a type of computer application. Facial recognition systems have been used more widely in robots and other types of technology. Systems for computerized facial recognition are classified as biometric since they analyze physiological traits of humans. Despite being less accurate than iris and fingerprint recognition systems in terms of biometric security, face recognition technology is nevertheless commonly used because of its non-contactless, In video surveillance and automatic picture indexing, facial recognition technologies have been used to assertain human-computer interaction.

The first face recognition algorithm was reported in the early 1970 by M. D. Kelly et al. (1971), and the first face recognition algorithm was created in the early 1970s. The advancement of computer and optical imaging technologies in the late 1980s led to the genuine start of the face recognition application phase in the late 1990s. Early face recognition research largely concentrates on geometry approaches, which pair straightforward characteristics with image processing strategies. Later, the holistic approach arose and gained popularity, including principal component analysis (PCA) and linear discriminant analysis (LDA). For matching all the local characteristics on a facial picture, a feature-based technique was then developed.

Deep learning is a subset of machine learning that uses artificial neural networks to mimic the learning process of human brain. Deep learning can achieve high accuracy when trained with large amounts of data. Convolutional Neural Networks (CNNs) is one of the deep learning algorithms and it is the most popular neural network model used for image classification problems. Since 2010, the ImageNet project have been running annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). In this challenge, software programs compete to correctly classify and detect objects and scenes. The challenge used a trimmed list of one thousand non-overlapping classes. The ImageNet project is a large visual database designed for use in visual object recognition software research. Recently, CNN models have achieved a good result in the ILSVRC competition, though large scales labelled data might be difficult to collect. Training deep neural network is time-consuming, as a result, a CNN based deep transfer learning for face recognition using small datasets is proposed. Transfer learning is a popular deep learning approach where the knowledge gained from a related task is transferred to a new task.GoogleNet is a type of convolutional neural network based on the Inception architecture. It utilizes Inception modules, which allow the network to choose between multiple convolutional filter sizes in each block. An Inception network stacks these modules on top of each other, with occasional max-pooling layers with stride 2 to halve the resolution of the grid. Compared with training deep neural networks from scratch.

**Problem Statement:** The use of a person's face in images, videos, or in real-time to recognize and verify their identification is known as facial recognition. Because FaceID is used to unlock iPhones, many people are familiar with facial recognition technology (however, this is only one application of face recognition). Typically, face recognition does not require a large collection of images to identify a person; rather, it merely detects one person as the device's owner and identifies them as such, denying access to others. In particular for purposes of presentation assault detection, face recognition can profit from the use of depth data obtained with inexpensive cameras. However, the output of these capture devices' depth images may include flaws like holes or general depth errors.

## Objective

In order to add a security component to the issue of facial recognition, this work compares the accuracy of enhanced (brightening the picture) and unenhanced photographs using transfer learning. The image is made brighter by using image enhancement techniques. GoogLeNet architectures and image processing in Matlab are used. The networks are contrasted with photos taken with a phone's camera

## Significant of study

The suggested approach can cut down on training time. Additionally, even with tiny data sets and great accuracy attained using GoogLeNet. This project is aimed to help improve security.

# CHAPTER 2

# LITERATURE REVIEW

## Introduction

This chapter reports on Transfer learning, Artificial Neural Network, convolutional neural network, data augmentation, related papers and summary.

## Transfer Learning

Transfer learning creates a model for one problem as the foundation for another related problem, transfer learning is a technique for machine learning. The use of pre-trained models as the foundation for computer vision and natural language processing tasks is a common deep learning strategy, given the limitless time and computing resources needed to develop neural network models for these problems as well as the huge gains in performance they provide on related problems, (Ramlee, 2020). For instance, knowledge obtained while learning to recognize vehicles may be applied to learning to recognize trucks. Transfer learning helps save time when working and improves the accuracy of a project. (Akhtar et al, 2017)

Two Typical methods employed in transfer learning include: Create a model approach and Pre-trained model approach that has been trained

**Create a model strategy**: It chooses a predictive modeling issue that is connected to it, and has a large amount of data, where the input, output, and/or ideas gained during the mapping from input to output all have some relationship. In confirming that any feature learning has taken place, the model must be superior to a naïve model. Then, a model on the second job of interest may be built from the model fit on the source task as a starting point. Depending on the modeling approach employed, this can include utilizing the entire model or only a portion of it. The model might perhaps need to be modified or refined based on the data for the task's accessible input-output pairs.(Singh & Prasad, 2018).

**Pre-trained Model Approach**: A model is selected as the source from a pool of models. Many research organizations publish models on sizable and difficult datasets, which may be part of the- selection of potential models. The model that has already been trained can then serve as the foundation for a model on the second job of interest. Depending on the modeling approach employed, this can include utilizing the entire model or only a portion of it. The model might need to be modified or improved based on the input-output pair data provided for the relevant job.

Two common examples of transfer learning with deep learning models are transfer learning with image data and transfer learning with language data. Transfer learning is frequently used to solve challenges in predictive modeling that take input from images. Transfer learning with image data; This might be a prediction challenge that requires the input of images or video (Tahir et al, 2021). A deep learning model that has been pre-trained for a difficult picture classification job like the ImageNet 1000-class photograph classification competition is frequently used for these kinds of tasks. The research organizations who create models for this competition, frequently make their final models available for reuse under a liberal license. On contemporary technology, these models may need days or weeks to train(Brownlee,2017).

These models are available for download and may be immediately integrated into new models that require picture data as input. The reason this strategy works so well is that the pictures were trained on a vast corpus of photographs and demand the model to make predictions on a variety of classes, necessitating the model to effectively learn to extract features from photographs in order to succeed on the challenge. Transfer learning with language data; with natural language processing issues that employ text as input or output, transfer learning is frequently used. A word embedding, which maps words to a high-dimensional continuous vector space where distinct words with identical meanings have similar vector representations, is employed for these kinds of challenges. These distributed word representations may be learned using efficient methods, and research groups frequently make pre-trained models available under permissive licenses once they have been trained on very large corpora of text texts (Brownlee,2017).

## Artificial Neural Network

The operation of organic nervous systems, such as the human brain, is a major source of inspiration for artificial neural networks (ANNs), which are computer processing systems. ANNs are primarily made up of a large number of interconnected computational nodes (also known as neurons), which collaborate to learn from the input in order to maximize the final output.

ANN may be used in a variety of applications, including color restoration, synchronizing a video's lip movement with an audio, automatic production of handwriting, automatic sound addition to silent films This will distribute it to the hidden layers from the input layer. The learning process is when the hidden layers consider judgments from the preceding layer and determine if a stochastic change inside itself worsens or improves the output. Deep learning is a term used to describe the stacking of many hidden layers (O’shea et al, 2015).

In tasks involving image processing, supervised and unsupervised learning are the two main learning paradigms.

Learning with pre-labeled inputs that serve as targets is known as supervised learning. There will be a set of input values (vectors) and one or more associated defined output values for each training example. Through accurate computation, this type of training seeks to lower the total classification error of the model of training example by training's output value.

Unsupervised learning differs from supervised learning in that there are no labels in the training set. The ability of the network to decrease or enhance an associated cost function often serves as a metric for success. It is crucial to remember that the majority of challenges for pattern recognition that focus on images often rely on categorization utilizing supervised learning. An ANN's three primary parts are the input layer, hidden layer, and output layer. It would load the input, frequently as a multidimensional vector (O’shea et al, 2015).

## Convolutional Neural Network

Similar to conventional ANNs, convolutional neural networks (CNNs) are made up of neurons that train to maximize their own performance. The fundamental building block of innumerable ANNs, each neuron will continue to take in input and carry out an action (such as a scalar product followed by a non-linear function). The whole network will still express a single perceptual scoring function from the input raw picture vectors to the class score at the end (the weight). The last layer will include loss functions related to the classes, and all of the standard techniques created for conventional ANNs are still applicable (Kizhevsky, 2012) The only significant distinction between CNNs and ANNs is that CNNs are more often employed in the field of picture pattern recognition. This enables us to further minimize the number of parameters needed to build up the model while encoding image-specific properties into the architecture, improving the network's suitability for image-focused tasks. Traditional ANN models frequently struggle with the computational complexity needed to calculate picture data, which is one of their biggest drawbacks (O’shea et al, 2015).

## Convolutional Neural Network (CNN) Architecture

CNNs place a lot of emphasis on the idea that the input will be made up of pictures. This concentrates the architecture's setup to best meet the requirements for handling the particular type of data. One of the main variations is that the layers of the CNN are made up of neurons arranged into three dimensions, the spatial dimensionality of the input (height and breadth) and the depth. The depth describes the third dimension of an activation volume rather than the total number of layers within the ANN.

In contrast to conventional ANNS, each layer's neurons will only link to a small portion of the layer before it. This would really mean, for the example, as we would have condensed the full input dimensionality into a smaller volume of class scores filed across the depth dimension, the input "volume" will have a dimensionality of 64 64 3 (height, width, and depth), leading to a final output layer composed of a dimensionality of 1 1 n (where n represents the possible number of classes). There are three different kinds of layers in CNNs. Convolutional, pooling, and fully-connected layers are what they are. A CNN architecture is created after these layers are layered. The fundamental capabilities of the various layer types The image's pixel values will be stored in the input layer. The convolutional layer will decide the output of neurons whose local input areas are related to those of those neurons the scalar product between their weights and the area related to the input volume is calculated.

The usage of learnable kernels is the main emphasis of the layer parameters. These kernels often have a low spatial dimension yet cover the whole depth of the input. Each filter is convolved across the spatial dimensions of the input by the convolutional layer as the data reaches it, creating a 2D activation map. The scalar product is calculated for each value in that kernel as we move through the input. When the receptive field size is set to 6 by 6 and the input to the network is an image with dimensions of 64 by 64 by 3 (an RGB-colored picture with 64 by 64 dimensions), for instance, In the convolutional layer, each neuron would have a total of 108 weights. (6 6 3), where 3 indicates the degree of connectedness inside the volume's depth. In order to put this into perspective, consider that a typical neuron seen in other types of ANN would have 12, 288 weights. Through the optimization of their output, convolutional layers are also able to considerably lower the model's complexity. The three hyperparameters of depth, stride, and zero-padding are used to optimize these.

After that, the pooling layer will simply down sample the input along the spatial dimension, thereby lowering the number of parameters in that activation. Pooling layers seek to gradually lower the representation's dimensionality and, as a result, further reduce the model's computational complexity and parameter count.

The "MAX" function is used by the pooling layer to scale the dimensionality of each activation map in the input. These typically take the shape of max-pooling layers with 2 2 dimensional kernels applied with a 2 stride along the input's spatial dimensions. This keeps the depth volume at its regular size while scaling the activation map down to 25% of its original size. There are only two commonly noted max-pooling strategies since the pooling layer is harmful. Typically, the pooling layers' stride and filters are both set to 2 2, meaning it will enable the layer to penetrate the input's whole spatial dimensions.

Additionally, overlapping pooling may be used when the stride and kernel sizes are both set to 2. A kernel size greater than 3 will typically cause the model's performance to suffer significantly because of the destructive nature of pooling. It's also crucial to realize that CNN designs may also include general-pooling in addition to max-pooling. Pooling neurons in general pooling layers may carry out a wide range of common operations, such as average pooling and L1/L2-normalization.

The fully-connected layers will next carry out the identical tasks as in conventional ANNs and make an effort to derive class scores from the activations, which can subsequently be used to classification. In addition, it is proposed that ReLu be applied in between these layers to enhance performance.

Neurons in the completely connected layer have direct connections to the neurons in the two adjacent layers; they are not linked to any neurons in those layers. This is comparable to how neurons are placed in conventional ANN models.

## Data Augmentation

Data augmentation is a group of methods for creating new data points from existing data in order to fictitiously enhance the amount of data. Making minor adjustments to the data or creating new data using deep learning models are examples of this. By applying multiple changes to photographs, data augmentation is a strategy that increases the quantity and diversity of training data, making images a viable solution to the limited-data face recognition problem (Data augmentation, 2020)

The performance and results of machine learning models may be enhanced by adding more data. Making straightforward modifications to visual data is common in data augmentation. In addition, fresh synthetic data is produced using generative adversarial networks (GANs). The following are typical image processing tasks for data augmentation: Resizing, flipping, cropping, rotating, padding, and many more operations are included in this approach. It makes the model robust with higher performance and aids in the resolution of issues like overfitting and data shortages. In essence, overfitting occurs when a network is unable to learn efficiently for a variety of reasons (Dilmegani, 2021).

Deep learning applications for machine learning in particular continue to diversify and grow quickly. In order to overcome the difficulties, the artificial intelligence field encounters, data-centric methods to model creation, such as data augmentation techniques, might be useful.

By creating fresh and diverse instances to train datasets, data augmentation is beneficial to enhance the performance and results of machine learning models. A machine learning model operates more effectively and correctly when the dataset is large and sufficient. Data collection and labeling may be time-consuming and expensive operations for machine learning models. Companies can lower these operating expenses by transforming datasets using data augmentation techniques. Data cleansing, which is required for models with high accuracy, is one of the phases in creating a data model.

However, the model cannot make accurate predictions for inputs from the actual world if cleaning decreases the representability of the data. By producing variables that the model could encounter in the real world, data augmentation approaches might help machine learning models become more resilient (Dilmegani, 2021). Benefits of data augmentation include the ability to forecast unusual occurrences, prevention of data privacy issues, and increased model prediction precision, lowering data labeling and collection costs, and reducing overfitting. (Dilmegani, 2021). If our models showed overfitting symptoms then, not only for our training dataset but also for our test and prediction sets, we can observe a decreased capacity to identify generic characteristics (Wikipedia, n.d.)

## Related papers

Using a neural network, (Al-Omari et al, 2009) demonstrated a method for identifying solitary digits. They created a technique for reading Arabic handwriting. They discovered a difficulty: Researchers have been working on handwriting number recognition for a very long time, notably in the last several years. Numerous areas of research deal with numbers, such as reading digits on bank checks or license plates. One such area is digit recognition.

A method for handling such an application may be a system for identifying solitary digits. In other words, to enable the computer to comprehend Arabic numerals entered manually by users and display them in accordance with computer-generated views. Engineers and scientists with expertise in image processing and different methods, including minimal distance, decision trees, and statistics, have been created using pattern recognition to address the issue of handwritten number recognition.

Effectively pick a segmentation technique that suits their needs. After successfully designing and implementing a neural network that can operate without being asked, the system was able to comprehend arabic numerals that were manually typed by users. The experiments used a total of 1300 isolated Arabic digits from 10 separate writers who supplied their handwriting, grouped into two data sets: 1000 digits are practiced, 300 digits are tested. On the test data set employed, their method achieved a 95% overall accuracy.

They came to the conclusion that neural networks appear to be superior to alternative methods for recognition. A face detector created and implemented in MATLAB by W. Mohsin et al. (2003) will find human faces in photos comparable to the training images. Numerous studies have been done on the issue of facial recognition. Color analysis, template matching, neural networks, support vector machines (SVM), maximum rejection classification, and model-based detection are just a few of the many approaches that have been employed. Designing algorithms that are effective for all illuminations, face colors, sizes, and geometries, as well as image backdrops, is challenging.

Face recognition is still a combination of art and science as a result. their approach employs categorization based on rejection. A group of weak classifiers that consecutively reject non-facial areas make up the face detector. Utilizing color segmentation, the non-skin color zones are first disregarded. A set following that, a variety of morphological processes are used to filter the excess noise produced by the preceding phase. The classification of the remaining linked areas is then done according to the geometry and the number of holes. The last step is to utilize template matching to find zero or more faces in each linked region. They came to the conclusion that because their method depends on the color information in the picture, it will not function with grayscale images.

The facial region of a person was discovered using a comparison between the input and output picture in the study using a novel algorithm for face detection technology that was constructed by segmenting the color image into RGB. The correctness of the face detection rate was 98%. Good results for the method's indications were produced by implementing the algorithm in the MATLAB program.

After analyzing the image and splitting it into three layers produced by the neural network that aids in detecting the features of the face, a new technique for identifying the face of color images was presented. This technique applies a classification technique to detect the face using the MATLAB program, and satisfactory results were obtained after using the basic equations. The algorithm was run on the photographs that did not have a mask, then another step was run on the ones that did. As for the last experiment, an algorithm was used using advanced and quick technology on a picture of a person wearing a mask and spectacles. Because the examples demonstrated that the new method is valid in the altered situation, the algorithm demonstrated its effectiveness.

(Suganthan et al, n.d) In the study, they suggest an effective tracking technique built on a convolutional neural network model that is quite basic. The suggested technique may concurrently identify the items from the background and obtain discriminant characteristics for visual tracking. This improves target location accuracy and is less unresponsive to fluctuations in appearance. The efficacy and resilience of the suggested approach are demonstrated by experimental results on sequences when compared to other state-of-the-art techniques. The visual tracking problem is addressed in this research in a discriminant way, using a simple convolutional neural network (CNN) to extract discriminant characteristics while also classifying the item from the backdrop. Ten difficult video sequences and five cutting-edge trackers are used in a thorough study to confirm the efficiency of the suggested strategy. They presented a novel method for broad object tracking using a fully convolutional neural network in the study. They undertake in-depth research on the characteristics of CNN features offline pre-trained on huge image data and classification job on ImageNet rather than using convolutional neural network (CNN) as a black-box feature extractor.

The tracking system's design was inspired by the discoveries. Convolutional layers at various levels were discovered to describe the object from various angles. While a lower layer carries more discriminative information and can more effectively distinguish the categories, a top layer stores more semantic traits and functions as a category detector. target protected against similar-looking distracters. During tracking, both layers are employed in tandem with a switch mechanism. It was also discovered that just a portion of neurons are important for a tracking target.

The development of a feature map selection approach can help decrease computation redundancy and boost tracking accuracy by removing noisy and pointless feature maps. A thorough analysis of the commonly used tracking benchmark reveals that the suggested technique performs significantly better than the state-of-the-art, and they experimentally provide certain crucial CNN feature qualities from the perspective of visual tracking. They suggested a tracking approach based on these qualities that makes use of fully convolutional networks that have already been trained for an image classification assignment. They noticed that convolutional layers at various levels have various qualities, and they jointly take these properties into consideration in order to assemble the target's semantic data and separate it from distracting background noise. To pick discriminative characteristics and exclude noisy or irrelevant ones, a principled feature map selection approach was further developed. They came to the conclusion that their strategy effectively enhanced tracking performance in difficult conditions.

Effects of contour enhancement on low-vision preference and visual search preferences for picture enhancement may be separated from search performance in individuals with vision impairment, according to (Woods et al, 2008).

The links between desire and performance for a specific range of mid-contrast sensitivity where an advantageous impact of enhancement may exist require further research. An method for infrared small-target identification was proposed by (Lui et al, n.d,) tiny-target improvement In contrast to deep learning-based and traditional algorithms, they from the standpoint of improving tiny targets in the input image, enhance the algorithm.

Sharpening spatial filters and upsampling are combined to improve this detection. Small targets are subtly improved by the sharpening spatial filter, making them more visible, more recognizable. The increased tiny targets are amplified by the upsampling process, creating point targets that are challenging to detect are generally simple to find. The suggested method successfully reduces the difficulty of detecting tiny items owing to their small size or gloom, doing in-depth comparisons with existing techniques using publicly available datasets.

Prior to these, Transfer learning in deep learning, Googlenet architecture, data augmentation and webcam was applied in this project.

# CHAPTER 3

# METHODOLOGY

## Introduction

The GoogleNet Architecture, transfer learning and training network are covered in this chapter. This chapter will show the evaluation of a face recognition system employing transfer learning methods and a pre-trained CNN Googlenet model.

## GoogleNet Architecture

Convolutional neural networks, such as GoogLeNet, are based on the inception design. It makes use of Inception modules, which provide the network the ability to select from a variety of convolutional filter sizes in each block. These modules are stacked on top of one another using an Inception network, occasionally max-pooling layers with stride 2 to reduce the grid's resolution in half.

The overall number of layers in GoogLeNet is 22, including 27 pooling layers and 9 inceptions modules layered linearly. The global average pooling layer is connected to the endpoints of the inceptions. The inception layer's goal is to cover a larger area while maintaining a high level of quality for the pictures' minute details. In order to get the most exact details, several sizes should be convolved simultaneously, starting with a smaller size (1x1) (5x5).

A sequence of gabor filters of various sizes is said to be able to manage the various scales of numerous objects better, thus having the benefit that every filter on the inception layer is learnable.

Adding additional layers and data is the simplest approach to boost deep learning performance; GoogleNet has 9 inception modules. The issue is that adding additional parameters makes your model more susceptible to overfitting. A parameter explosion on the inception layers is therefore prevented by all

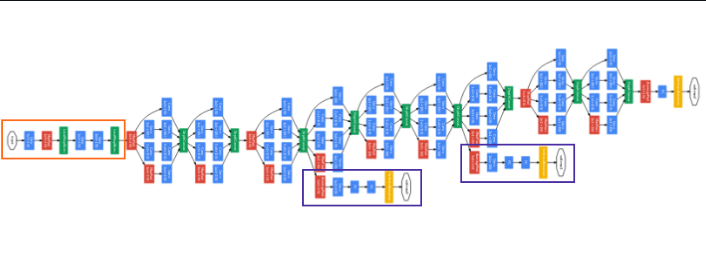
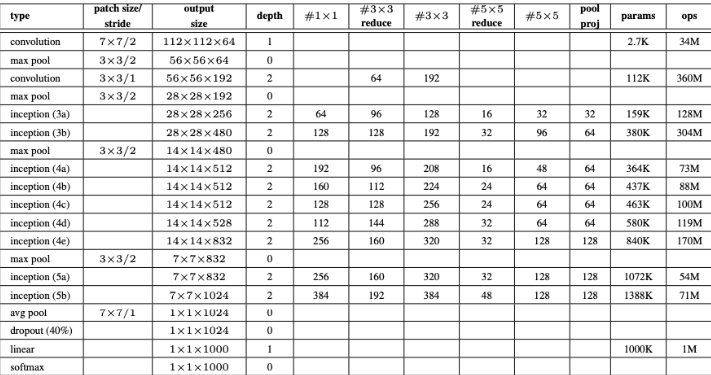


Figure 1.0: Representation of full Google net architecture (geeksforgeeks, 2021)

Table 1.0 : Detailed architecture and parameters (geeksforgeeks, 2021)



## Transfer learning and training Process.

## Dataset

Dataset is a collection of related sets of information that is made up of separate elements but can be manipulated as a unit by a computer. In this work, there are 370 files in 20 folders in the dataset. Each folder had a maximum of 30 separate pictures of the same person taken at various angles and positions. A total of 70% of the files were utilized for training, 20% for testing, and 10% for validation.

Table 2.0 : Dataset specification and Values

|  |  |
| --- | --- |
| **DATASET SPECIFICATION** | **VALUES** |
| Number of files | 370 |
| **File Type** | **Joint Photographic Expert Groups** (JPG) |
| **Camera** | Mobile phone: Tecno cammon 16 |
| **Image Size** | 76.4 KB – 3.0 MB |

## 2. Importing and Loading Data

In this section images in the folders are imported and loaded in the pre trained system as a data. In using the new photos as an image datastore, They were initially unzip. The photos are automatically labeled by imageDatastore based on the folder names, and the information was saved as an ImageDatastore object. In order to train a convolutional neural network, ton of image data were stored and read large batches of images quickly utilizing an image datastore. Figure 2.9 shows how images were imported in Matlab

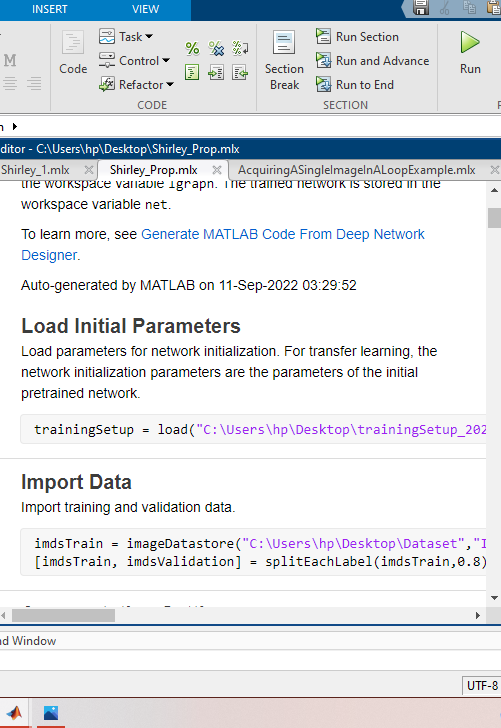


Figure 2.9: Screen shot of Matlab codes for importing and loading data

The script was ran to load training and validation data, build the network layers, and train the network. The workspace variable lgraph contains a list of the network layers. The workspace variable net contains a copy of the trained network.

## 3. Loading Pre-trained Network

Pre-trained network refers to a saved network created previously by someone and trained on a huge dataset to solve a similar problem. To load this network, Deep Learning ToolboxTM Model for GoogLeNet Network to load the pre-trained GoogLeNet neural network was installed on HP Pavillion, 16 GB of RAM, and a Windows platform with an Intel Core i3- CPU operating at 2.7 GHz. DeepNetworkDesigner was used to provide an interactive representation of the network design and thorough details about the network layers.

deepNetworkDesigner(net).

The input pictures for the first layer, which is the image input layer, must be 224 x224 x3, where 3 is the number of color channels.

net for inputSize.

Layers(1).

The input size is inputSize = 13 224 224 3

The figure 3.0 shows the interface of loading the network.

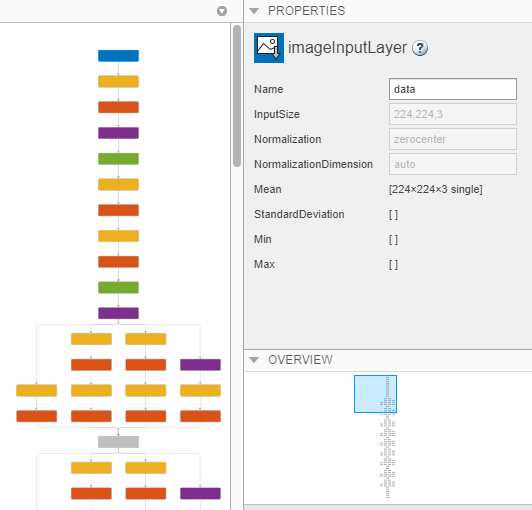


Figure 3.0: Interface for loading pretrained network

## 4. Replacing Final Layers or Adding layer Branches

The pre-trained network net's fully connected layer and classification layer were set up for 1000 classes. The knowledge on how to integrate the features that the network extracts into class probabilities, a loss value, and predicted labels is contained in these two layers, loss3-classifier and output in GooLeNet. These two layers were swapped out by new ones that were tailored to the fresh data set in order to retrain a pretrained network to categorize new photos.

From the trained network, the layer graph was extracted using the following Matlab code.

layerGraph(net) = lgraph;

A new completely connected layer that has the same number of outputs as classes was added to replace the previous layer that was fully linked. Increase the WeightLearnRateFactor and BiasLearnRateFactor values of the completely linked layers to make learning in the new layers quicker than in the transferred layers.

The figure 4.0 displays the interface for adding layers on the network.

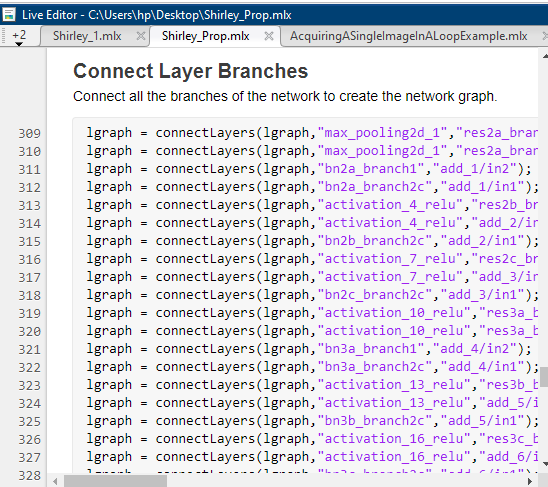
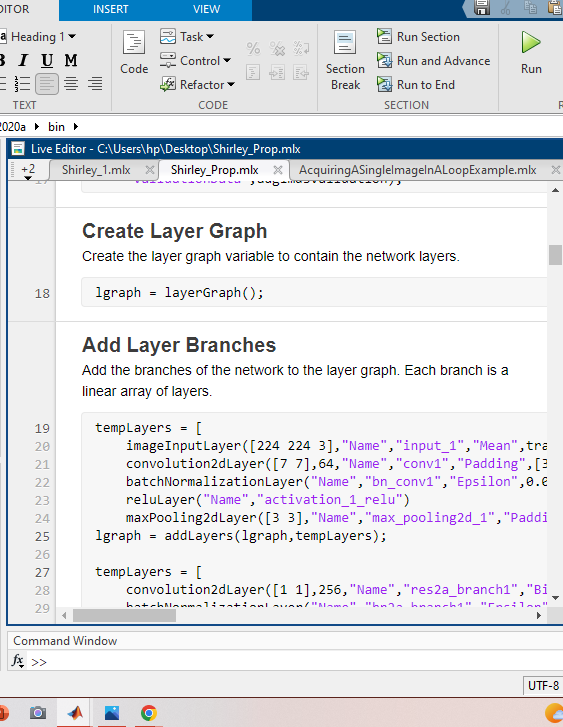


Figure 4.0: Adding layers and Extracting layer graph

## 5. Training Network and Augmentation of Data

The picture datastores include photos of various sizes, however the network requires input images of size 224 x224 x3. Training pictures were automatically resized using an enhanced image datastore. The training pictures went through the following extra augmentation operations: randomly flip the training images along the vertical axis, and randomly translate them up to 30 pixels in both the horizontal and vertical directions. The network was then trained by pressing (RUN) in MatLab. Validation pictures were automatically resized without doing additional data augmentation. HP Pavilion, 16 GB of RAM, and a Windows platform with an Intel Core i3- CPU operating at 2.7 GHz were utilized for the training. The MATLAB 2020a tool was used to complete the tasks of method assessment, feature selection, and classification.

The figure 2 is the interface for training and augmentation of the dataset.

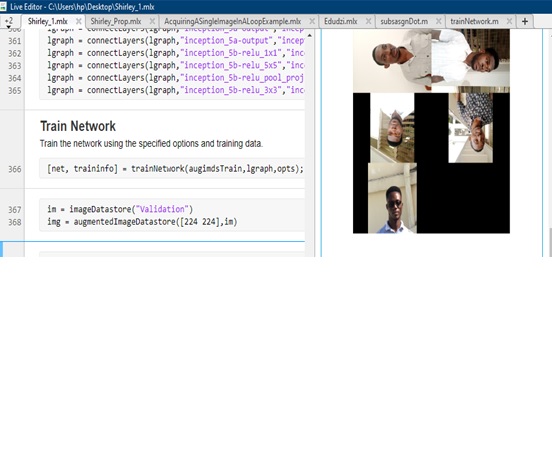


Figure 2: Training and Augmentation Section

By applying numerous changes to photos, data augmentation is a strategy that increases the quantity and diversity of training data, making them a viable solution to the challenge of face recognition with limited data. Web camera was loaded to aid in live assessment

# CHAPTER 4

# RESULTS AND DISCUSSION

## Introduction

This chapter reports on the results from the training process graph and the HP Pavilion webcam.

## RESULTS AND DISCUSSION

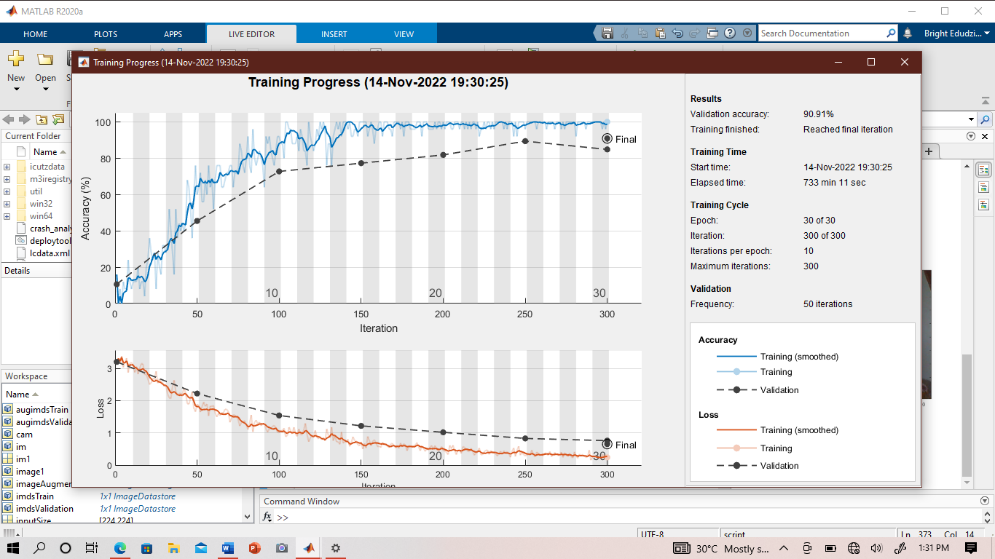


Figure 3: Training Progress Graph for Accuracy and Loss

From the training progress plot, overfit model is easily diagnosed by monitoring the training accuracy, validation accuracy, training loss, validation loss by evaluating on both training and validation set. If the model is overfitting, the training accuracy will greater than the validation accuracy while the validation loss value will be greater than the training loss value.

From Figure 3, the graphical representation of the training progress for accuracy and loss, it could be seen that the validation accuracy is 90.91 percent which implies that overfitting did not take place during the training process. Also from the training cycle there was 30 of 30 epoch, 300 of 300 iteration, with 50 validation iteration. From the accuracy graph in Figure 3, the thick blue line in the graph shows how smooth it was during training progress, the light blue line indicates the training, and the dotted black line indicates the validation, From the loss graph in Figure 3, the thick red line in the graph shows how smooth it was during training progress, the light red line indicates the training, and the dotted black line indicates the validation

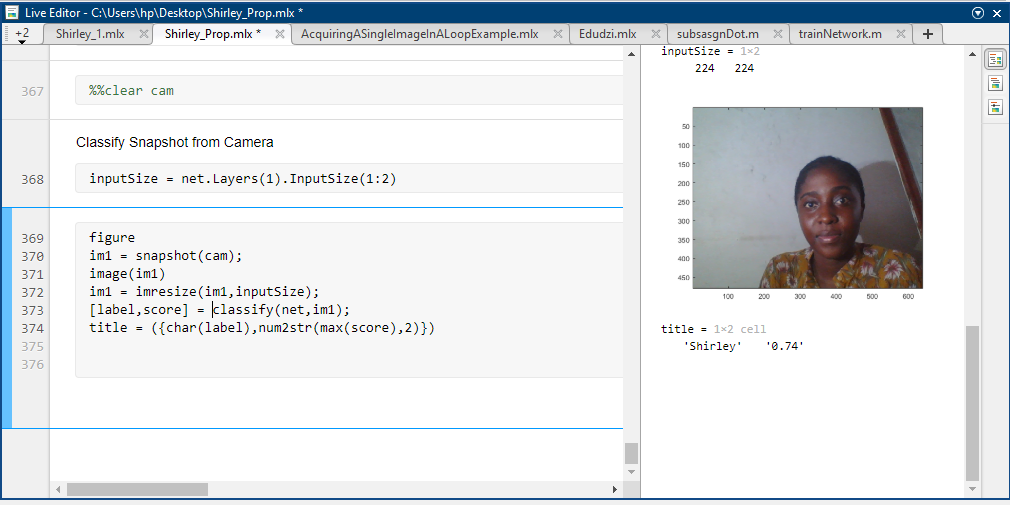


Figure 4: Results from webcam snapshot

From Figure 4, it could be seen that the live camera (webcam), was able identify the individual that is Shirley with 0.74 or 74% accuracy. The percentage it indicates, shows how valid or true it has identify the individual, the higher the percentage the higher it has identified the individual and vice versa. For example if the system identifies an individual with 0.06%, that shows how low it was able to identify the person, hence it would give a wrong prediction.

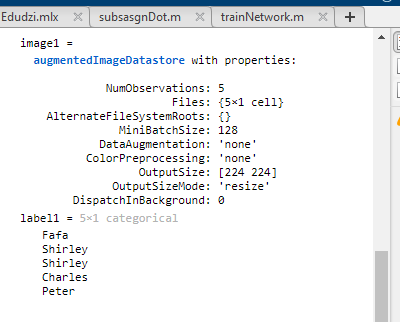
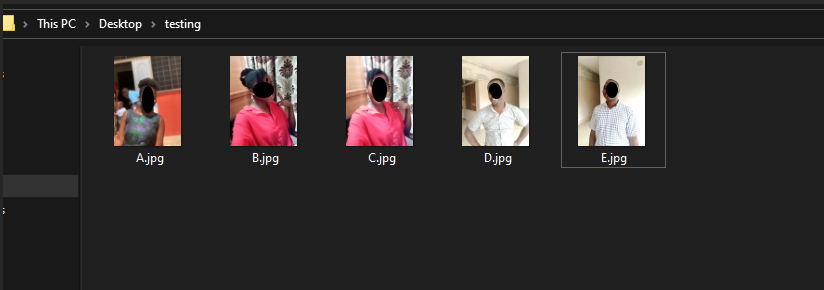
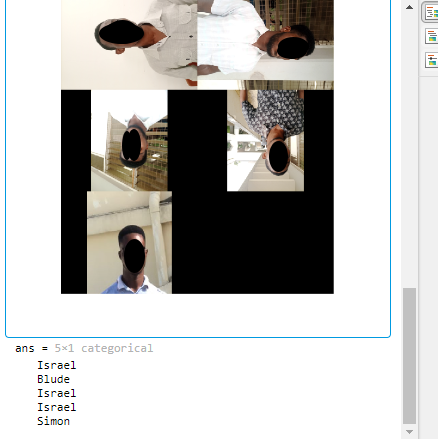


Figure 5: Results from Augmentation and Testing

Figure 5 Displays the results from the Matlab augmentation section, where it identifies images that have been flipped and rotated, from the results; it was able to correctly identify three images with its respective folder name (Israel, Blude and Simon) out of five images, Some of these images where repeated out of the five and the system wasn’t able to identify. Also display of results from a testing folder, where the system identifies images in the folder from the desktop.

Table 3.0 : A table showing wrong and correct prediction from testing result.

|  |  |  |
| --- | --- | --- |
| **CLASS PREDICTED** | **WRONG PREDICTION** | **CORRECT PREDICTION** |
| Israel  Simon  Blude  Agyabony  Shirley  Charles  Fafa  Peter | 1  0  0  2  0  0  0  1 | 2  1  1  0  2  1  1  1 |

From the table 3.0, the numbers allocated for the wrong and correct predictions shows the number of times it was accurate in predicting the names from a group of different pictures, out of the 90.91% accuracy, errors may as a result of the remaining 9.09%, since the accuracy of the system was not 100%.

# CHAPTER 5

# CONCLUSION AND RECOMMENDATION

## CONCLUSION

In this project, a face-recognition technique based on tiny data sets and pre-trained CNN model from GoogleNet is implemented. Data augmentation is used because deep learning requires a modest amount of data for training. This can protect against overfitting and improve the model's functionality. Results from the Progressive graph showed how accurate 90.91% the system was with 9.09% error, from the testing section we could see the errors where the system was unable to identify images according to its class or name

## RECOMMENDATION

Other deeper pre-trained CNN models, like ResNet, DenseNet, and VGG-Net, can be used for the transfer learning model. Enhancement of the images before training may contribute to its accuracy and effectiveness. These models can then be compared and evaluated to see which one can provide the best performance and st

# REFERENCES

Alex, A. IBM Developer skills network, Deep learning fundamentals with keras, *“Video”.*

Trigueros, D. S., Meng L., & Margeret H. (2018). (PDF) Face Recognition: From Traditional to Deep Learning Methods*, ResearchGate*.

Winarno, E. (2018). Multi-View Faces Detection Using Viola-Jones Method: *Journal of Physics: Conference Series J.Phys*. Conf. Ser 1114 012068

Shaker, F., & Abdulelah, Z. (2020). Face Detection By some Methods based on MATLAB: *Journal of Al- Qadisiyah for Computer Science and Mathematics* 12 (4) pp 12–17

Keiron, O., & Ryan, N. An Introduction to Convolutional Neurals: *School of Computing and Communications*. Lancaster University, Lancashire, LA1 4YW:

Krizhevsky, R., Alex, P., Ilya, S., & Geoffrey, E. (2021) "ImageNet Classification with Deep Convolutional Neural Networks." Advances in neural information processing systems (25).

Liu, S., Chen, P., & Wo ´zniak, M. Image Enhancement-Based Detection with Small Infrared Targets .

Ejaz, S., Islam, M., ifatullah, M., & Sarker, A. (2019) Implementation of principal component analysis on masked and non-masked face recognition 1 st *International Conference on Advances in Science Engineering and Robotics Technology* *(ICASERT)* pp 1–5

Liao, M., & Gu, X. (2020). “Face recognition approach by subspace extended sparse representation and discriminative feature learning,” Neurocomputing, vol. 373, pp. 35–49.

Minaee, S., Abdolrashidi, A., & Wang,Y. (2017). "Face recognition using scattering convolutional network," *IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, Philadelphia, PA, pp. 1-6, doi: 10.1109/SPMB.2017.8257025.

Singh, S., & Prasad, S. (2018). “Techniques and Challenges of Face Recognition: *A Critical Review,” Procedia Computer Science*, vol. 143, pp. 536–543.

Satgunam, P., Woods, L., Luo, G., Bronstad, P., Reynolds, Z., Ramachandra, C., & Peli, E. Effects of contour enhancement on low-vision preference and visual search. *Optom Vis Sci.*

Simard, P.Y., Steinkraus, D., &Platt, J.C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In: null. p. 958. IEEE .

Szegedy, W., Christian, P., Wei, L., Yangqing, J., Pierre, S., Scott, R., Dragomir, A., Dumitru, E., Vincent, V., & Andrew, R. (2015) "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition.

Karras, T., Laine, S., & Aila, T. (2019). A Style Based Generator Architecture for Generative Adversarial Networks IEEE/CVF *Conference on Computer Vision and Pattern Recognition (CVPR)* Long Beach CA USA pp 4396- 4405 doi: 10.1109/CVPR.2019.00453

Akhtar, K. & Rattani, A. (2017) “A Face in any Form: New Challenges and Opportunities for Face Recognition Technology,” Computer, vol. 50, no. 4, pp. 80–90.