

Person recognition system using CNN & Transfer Learning

Rajesh Raj Tudu



Department of Electrical Engineering
National institute of Technology, Rourkela

Person Recognition system using CNN & Transfer Learning

*Dissertation submitted in partial fulfilment
of the requirements for the degree of*

Bachelor of technology

in

Electrical Engineering

by

Rajesh Raj Tudu

(Roll Number: 120EE0521)

*based on research carried out
under the supervision of*

Prof. Dipti Patra



13 May 2024

Department of Electrical Engineering
National Institute of Technology, Rourkela



Department of Electrical Engineering
National Institute of Technology, Rourkela

May 13, 2024


Certificate of Examination

Roll Number: *120EE0521*

Name: *Rajesh Raj Tudu*

Title of Dissertation: *Person recognition system using CNN and Transfer Learning*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfilment of the requirements of the Bachelor of Technology in Electrical Engineering at National Institute of Technology Rourkela. We are satisfied with the volume, quality, correctness, and originality of the work.


Prof. Dipti Patra 13/5/2024
Principle Supervisor



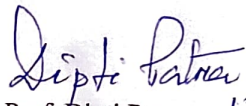
Department of Electrical Engineering
National Institute of Technology, Rourkela

Prof. Dipti Patra
Professor

May 13, 2024

Supervisor's Certificate

This is to certify that the work presented in the dissertation entitled *Person recognition system using CNN and Transfer Learning* submitted by *Rajesh Raj Tudu*, Roll Number: *120EE0521*, is a record of original research carried out by him under our supervision and guidance in partial fulfilment of the requirements of the degree in *Bachelor of Technology in Electrical Engineering*. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.


Prof. Dipti Patra
Principle Supervisor

13/5/2024

Dedication

I dedicate this to my Mother , A super woman for me who take care of me and my elder sister after my father.

A handwritten signature in blue ink, appearing to read 'Rajesh', with a long horizontal flourish extending to the right.


Signature

Declaration of Originality

I, *Rajesh Raj Tudu*, Roll Number *120EE0521* hereby declare that this dissertation entitled *Person recognition system using CNN and Transfer Learning* presents my original work carried out as an Engineering of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

May 13, 2024
NIT Rourkela


Rajesh Raj Tudu

Acknowledgement

My profound appreciation is given to Prof. Dipti Patra of the National Institute of Technology, Rourkela, for her great advice and steadfast assistance during the creation of the Person Recognition System employing CNN and Transfer Learning. Her knowledge and guidance have been invaluable in helping to shape our project. Her sage advice and encouragement have also been tremendously appreciated, and they have improved my learning process. Without her direction, this project would not have been feasible, and I sincerely appreciate the chance to work under her direction.

May 13 ,2024
NIT Rourkela



Rajesh Raj Tudu
Roll Number: 120EE0521

Content Table

1	Abstract	
2	Introduction.....	10
3	Motivation.....	10
3.1	Research challenges.....	10
4	Literature Review.....	11
5	Related Works.....	12
5.1	Convolutional Neural Network (CNN).....	12
5.1.1	Convolutional Layer.....	13
5.1.2	Pooling Layer.....	14
5.1.3	Full Connected Layer.....	16
5.2	Transfer Learning.....	16
5.3	Pre-trained CNN.....	18
6	Dataset.....	19
6.1	Facial Datasets.....	19
6.1.1	The Extended Yale Face Database B(Cropped).....	19
6.1.2	The Extended Cohn-Kanade Dataset (CK+).....	20
7	Methodology.....	20
7.1	MATLAB code.....	20
7.2	Approch-1.....	23
7.3	Approch-2.....	23
7.4	Learning rate.....	24
7.5	Dropout.....	25
8	Work Results.....	26
8.1	The cross-entropy result.....	26
8.2	Accuracy result.....	26
8.3	The computational time results.....	27
9	Conclusion.....	27
10	Future Scope.....	28
11	Acknowledgment.....	28
12	Reference.....	28

1 Abstract

Identifying a person or developing a person recognition system is one of the most challenging fields in computer vision. A key success factor in this field is the extraction of large amounts of data and classification of input images using deep learning algorithms. The application of deep learning algorithms to extract the most relevant features from an image and classify them into different categories produces outstanding results. By utilizing previously available data knowledge, CNN and Transfer learning algorithms assist in achieving the desired accuracy. As a result of transfer learning, previously acquired knowledge is utilized and training time is reduced.

As deep learning and related technologies advance, the field of person recognition in computer vision is rapidly evolving. As a result of the use of CNNs, transfer learning, and other deep learning techniques, researchers and developers are making significant advances in the ability to accurately identify individuals in images, which have broad implications for security, surveillance, and personalized customer service.

2 Introduction

Deep Learning (DL) is a subset of machine learning methodologies based on neural network representations to facilitate pattern recognition. Since its inception in the late 20th century, DL has evolved significantly within the broader area of computer vision. An artificial neural network is based on the workings of the human brain. It mimics the behaviour of neurons by gathering data from visual inputs and categorizing them, thus aiding in the identification process. There are three primary stages of recognition within DL: Detection, Extraction, and Recognition.

- 1) Detection is the first step where crucial components are identified. As an example, in the context of person identification, this phase is concerned with identifying the face, including the nose, eyes, mouth, and overall structure of the head.
- 2) The Extraction stage is the next step. Primitive edges or features are extracted from the input data and converted into a computer-readable format. At this stage, Holistic template-based techniques and geographic features-based techniques, along with Convolutional Neural Networks (CNNs), are used to generate a feature vector. When it comes to facial recognition, for example, this stage involves extracting primitive features that are unique to each individual face.
- 3) In the Recognition or Classification stage, the classification of the object or face is determined. The purpose of this stage is to identify and distinguish between different objects or faces through intricate classification procedures.

As a subset of machine learning, deep learning has revolutionized various fields, particularly computer vision. The use of neural network representations inspired by the human brain allows DL models to effectively recognize patterns, enabling applications ranging from image classification to natural language processing. By combining the three major phases of detection, extraction, and recognition, DL has significantly advanced the field of computer vision, providing innovative solutions to complex recognition problems.

3 Motivation

The use of Transfer Learning and Convolutional Neural Networks (CNNs) to person recognition is a significant advancement in artificial intelligence with far-reaching implications for security, surveillance, and tailored services, among other areas. By utilizing deep learning's powerful capabilities, this novel system demonstrates unmatched ability to identify people based on their unique characteristics, such as facial features or walking patterns, marking a revolutionary change in how we engage with technology and each other.

Convolutional Neural Networks, a subclass of deep neural networks specifically designed to handle visual input with exceptional efficiency, are at the core of this technological innovation. These networks are excellent at finding complex patterns and characteristics in pictures, which makes them perfect for applications like facial recognition. Through the use of CNNs' hierarchical structure, the system is able to extract complex information from unprocessed visual inputs, which allows it to identify minute differences in face features across different people.

Furthermore, the effectiveness of this ground-breaking method of person recognition is further increased by the addition of Transfer Learning. By using transfer learning, the model may benefit from the information stored in neural networks that have already been trained on large datasets for common image identification tasks. This knowledge transfer speeds up the implementation of reliable person

identification systems in real-world situations by enabling the system to quickly adapt to new recognition tasks with low data needs.

The effects of this technology are wide-ranging. Accurate person identification has the potential to improve public safety in the fields of security and surveillance by quickly identifying criminals or possible threats in crowded areas. Moreover, accurate person identification can enable customized experiences in personalized services like retail or entertainment by providing people with recommendations or services that are specific to their particular tastes and behaviour.

Convolutional neural networks and transfer learning used to recognize faces, in essence, mark a turning point in artificial intelligence that has the potential to reshape the nature of human-technology interaction and spur revolutionary developments in a wide range of fields. The influence that this technology will have on society as it develops will be nothing short of transformative, influencing a future in which the lines separating humans from machines will increasingly blur.

3.1 Research Challenges

There are several problems or challenges that face during face recognition while using deep learning Algorithms.

- Deep learning models represent a significant problem in person recognition because of the wide range of lighting situations, stances, and facial emotions. This unpredictability presents a major challenge for the profession and necessitates careful consideration in research projects. In order to guarantee dependable and accurate person recognition, researchers struggle to develop approaches that can withstand these various situations.
- The difficulty of identifying people in the face of changing environmental conditions emphasizes the significance of this study area. Innovative methods that can reliably record and decipher face characteristics in a variety of lighting conditions, stances, and emotions are needed to tackle this difficulty. As a result, researchers working to advance this technology continue to focus on improving the flexibility and resilience of deep learning models in face recognition.

The flexibility of CNNs and Transfer Learning to various settings and circumstances is a major benefit for person recognition. They are perfect for usage in a broad range of real-world contexts since they can recognize people under varied lighting conditions, numerous camera angles, and variations in face expressions.

4 Literature Review

In this section the related worked done on the field of face recognition are discussed.

- A person's face is the main characteristic that makes identification easier. Faces vary, even in identical twins. Therefore, facial recognition and identification are required for people to recognize one another. A biometric method is used by a facial recognition system to confirm an individual's identification. These days, facial recognition is a widely utilized technology in a wide range of applications, such as phone unlocking systems, home security systems, and criminal identification systems. This method is more secure because it only needs a photograph of your face, instead of a key or card. A person recognition system usually has two phases: face detection and face identification. This paper introduces the concept of developing and optimizing a deep learning-based facial recognition system.

- This study uses the robust regression technique, which is commonly used to address object recognition problems influenced by illumination factors. The issue that needs to be resolved is how well face recognition works in various lighting situations. The Yale Face Database B Cropped, one of the face databases, was used to test each lighting condition of the face data. The Yale Face Database B Cropped contains ten individuals, each of whom has 64 illumination-based image variations that are further divided into five contrast-level subsets. In experiments, all subsets are usually used as training data, and the remaining data are used as testing data. The outcomes show a general face recognition performance in different lighting settings.

5 Relative Works

A detailed discussion of the main algorithms used to accomplish the tasks is included in this section, such as Convolutional Neural Networks and Transfer Learning.

5.1 Convolutional Neural Network (CNN)

A specific type of Deep Learning algorithms called Convolutional Neural Networks (CNNs) is used to extract complex information from input photos. In a CNN, layers upon layers of neurons function sequentially, with each layer using linear transformations to enable nonlinear operations on the outputs of the one before it. CNNs are primarily composed of convolutional and pooling layers. The convolutional layers use trainable weights, whereas the pooling layers use fixed functions to produce the appropriate activations. The neural network design resembling a grid works well when processing data with predetermined characteristics.

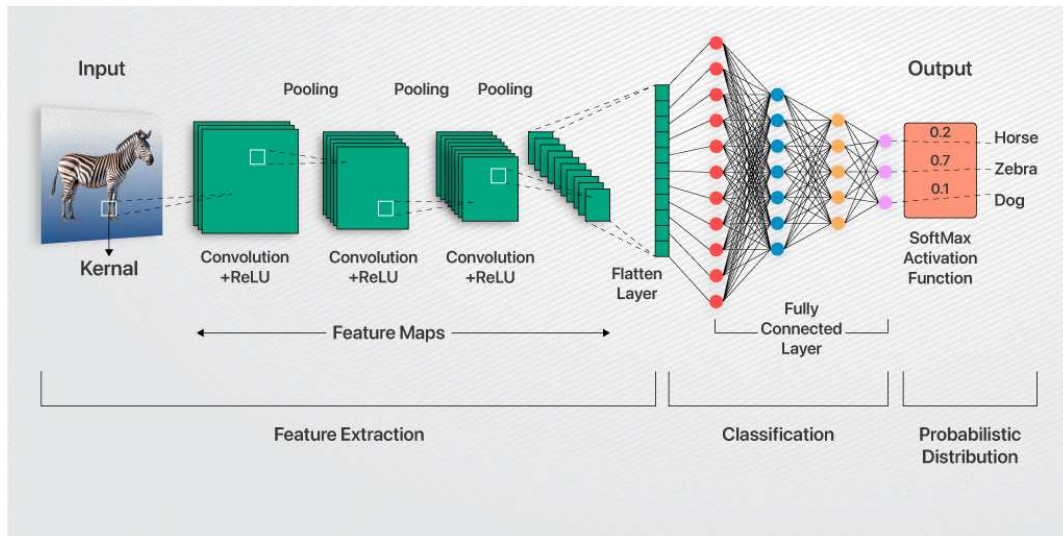


Figure-1: CNN working

CNNs are now recognized as very effective systems for pattern identification and have become a popular model for image classification jobs throughout time. The efficacy of the model is attributed to its capacity to identify complex patterns in pictures, hence permitting precise categorization. Preprocessing is done on input picture data before it is sent into the network. Because it directly affects layer calculations later on, it is imperative that all input pictures have consistent dimensions that is, similar height, width, and channels.

CNNs are important because they can automatically extract relevant information from pictures, which makes it easier to recognize patterns. CNNs improve their capacity to recognize intricate patterns and variances in picture collections by iteratively honing learnt representations through layers. Their versatility renders them very appropriate for an extensive range of uses, including semantic segmentation, object identification, and picture classification.

CNNs are essentially a mainstay in the field of computer vision, providing a strong foundation for addressing various image processing applications. CNNs are expected to further transform a number of disciplines as research and development efforts continue, spurring breakthroughs in industries including healthcare, autonomous cars, and surveillance systems. Basically, CNN is composed of three main layers, namely:

5.1.1 Convolutional layer

The convolutional layer is the main component of a CNN, responsible for most computation. It comprises input data, a filter, and a feature map. The input is a 3D colour image with dimensions of height, width, and depth, representing RGB. A feature detector, also known as a kernel or filter, moves across the image's receptive fields to check if a feature is present, known as convolution.

The feature detector is a two-dimensional array of weights representing an image part. The filter size is typically a 3x3 matrix, determining the size of the receptive field. The filter is applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is fed into an output array, and the filter shifts by a stride until the kernel has swept across the entire image. The final output is a feature map, activation map, or convolved feature.

The feature detector maintains fixed weights across images, known as parameter sharing. Some parameters, like weight values, adjust during training through backpropagation and gradient descent. However, three hyperparameters, affecting output volume, must be set before neural network training begins. The depth of the output is influenced by the number of filters used, with a depth of three being achieved with three distinct filters.

The stride, the distance the kernel moves over the input matrix, is a key factor, with larger values resulting in smaller outputs. Zero-padding is typically used when filters don't fit the input image, setting all elements outside the matrix to zero. here are three types of padding in a convolution: valid padding, same padding, and full padding. Valid padding drops the last convolution if dimensions don't align, same padding ensures the output layer's size is the same as the input layer, and full padding increases the output's size by adding zeros to the input border.

$$A = I * F$$

Where A is the Activation function or Features map, I is the Input image, F applied filters.

In person recognition system using a convolutional layer is to detect certain attributes associated with humans from input photos. For the sake of clarity, we will simplify the example by assuming grayscale photos.

A picture with dimensions of 32 × 32 pixels. The input picture is a representation of a person's lips, nose, and eyes. To extract pertinent characteristics, we'll employ a single 3 × 3 pixel filter. The Input image I with matrix of 32 × 32 dimensions. The filter F with 3 × 3 dimensions.

The convolution operation involves sliding the filter over the input image and computing the dot product at each position.

$$A_{i,j} = \sum_{m=0}^2 \sum_{n=0}^2 I_{i+m,j+n} \times F_{m,n}$$

$A_{i,j}$ is the value at position (i,j) in the output features map, $I_{i+m,j+n}$ represent the pixel value in the input image at position $(i+m,j+n)$, and $F_{m,n}$ is the corresponding weight in the filter at position (m,n) .

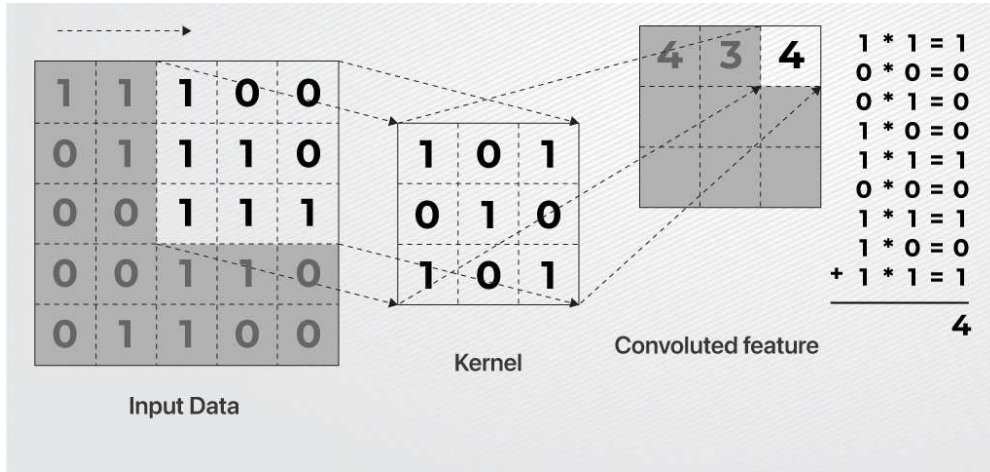


Figure-2: Convolution Operation

In given example 0 means dark and 1 means white colour. By extracting pertinent characteristics from the input image for later processing in the person recognition system, this procedure creates the output feature map A.

5.1.2 Pooling Layer

In Convolutional Neural Networks (CNNs), pooling layers—also called down sampling—are essential because they reduce the dimensionality and number of parameters in the input data. Pooling layers function without weights, in contrast to convolutional layers, which use trainable filters. Alternatively, they make use of a filter that runs through the whole input and applies an aggregate function to values that fall inside its receptive field. The output array that is produced at the end of this operation effectively reduces the input's spatial dimensions.

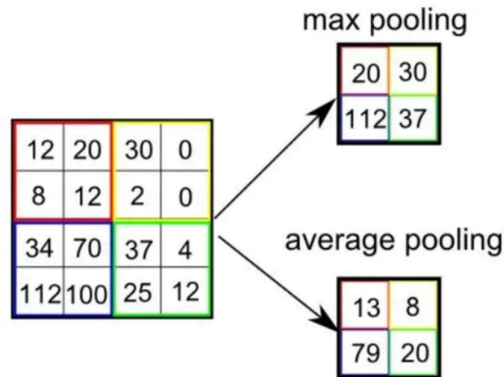


Figure-3: Pooling Operation

It helps to combine data while maintaining the important characteristics that were taken out by the convolutional layers that came before them. These layers reduce the danger of overfitting and enhance computational efficiency by summarizing local information using aggregation functions such as max pooling or average pooling. Moreover, down sampling strengthens the network's resistance to spatial translations by guaranteeing that critical characteristics are retained at different points in the input data. In general, pooling layers are essential parts of CNN designs that help with efficient dimensionality reduction and feature extraction in image processing applications. There are two types of Pooling:

1) Max pooling: A key function of convolutional neural networks (CNNs) is max pooling, which is used to downsample feature maps and minimize their spatial dimensions while keeping relevant features. A filter is slid across the input feature map during max pooling, usually in non-overlapping areas, and the maximum value is extracted for each zone. As a typical overview of the traits found in that area, this maximum value is used. It efficiently emphasizes the most prominent features found by earlier convolutional layers while removing less important data by choosing the maximum value. This procedure helps to maintain important properties for layers that come after while lowering memory use and computational complexity.

Because the maximum value inside each zone remains constant despite slight changes in the input, max pooling also offers a degree of translation invariance. This characteristic improves the model's resistance to changes in input picture quality by assisting the network in concentrating on the existence or lack of features rather than their exact positions.

All things considered, max pooling is essential to CNN designs because it makes dimensionality reduction easier, boosts computational performance, and encourages translation-invariant feature extraction, all of which help the network acquire meaningful representations from input data.

2) Average pooling: Convolutional neural networks (CNNs) use average pooling, a crucial function, to downsample feature maps while keeping important information intact. A filter traverses the input feature map in non-overlapping sections during average pooling, determining the average value within each region.

In contrast to maximum pooling, which chooses the highest value, average pooling calculates the mean value, offering a broader overview of the characteristics found in the area. Average pooling lowers the chance of excluding potentially significant information from the feature map by averaging the values to provide a smoother downsampled representation of the input.

The capacity of average pooling to create a translation invariance comparable to max pooling is one of its benefits. Average pooling improves the model's resilience to spatial fluctuations in the input by assisting the network in focusing on the general existence of features rather than their exact positions. This is achieved by calculating the average value within each area.

All things considered, average pooling benefits CNN architectures through dimensionality reduction, computational efficiency, and translation-invariant feature extraction encouragement. Because of its well-balanced downsampling strategy, the network can effectively learn representations from input data while preserving relevant information and shrinking the spatial dimensions of feature maps.

5.1.3 Fully connected (FC) layer

An essential element of neural network topologies, such as convolutional neural networks (CNNs), is the fully connected layer, often referred to as the dense layer. Every neuron in this layer is linked to every other neuron in the layer above, creating a complex web of connections.

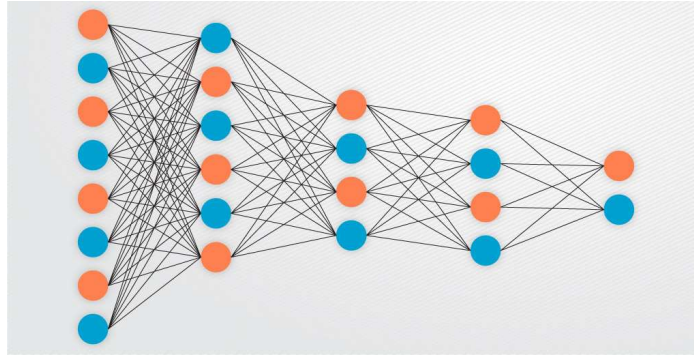


Figure-4: Fully Connected layer

A bias term is introduced to each neuron during forward propagation, and the input from the preceding layer is multiplied by a weight matrix. This procedure adds nonlinearity to the network by first computing a weighted sum of the input characteristics and then applying an activation function. Tanh, sigmoid, and ReLU (Rectified Linear Unit) are examples of common activation functions.

Output shows the characteristics or predictions made by the network that are pertinent to the current job. In order to acquire probability distributions over the classes, the output of classification tasks is frequently run through a softmax function. The result of regression tasks is immediately used to the prediction of continuous variables.

The network may discover sophisticated patterns and correlations within the data because to the fully connected layer's facilitation of complex mappings between input and output data. The fully connected layer is essential to neural network topologies because it can capture high-level representations and enable precise predictions, even with its computational complexity and propensity for overfitting.

5.2 Transfer Learning

In machine learning and deep learning, transfer learning is a potent strategy that entails using information from one task to enhance performance on a related one. Transfer learning reduces the need for large amounts of labelled data and computing resources by enabling pre-trained models to be repurposed for other tasks, in contrast to standard machine learning techniques where models are trained from scratch on a given dataset.

Fundamentally, transfer learning makes use of the notion that information gained from resolving one issue may help with other issues. The transfer of acquired characteristics or representations from the source task to the target task is how this knowledge transfer takes place. In actual use, this typically entails fine-tuning a previously trained model using a fresh dataset or task, changing the model's parameters to better fit the unique features of the fresh data.

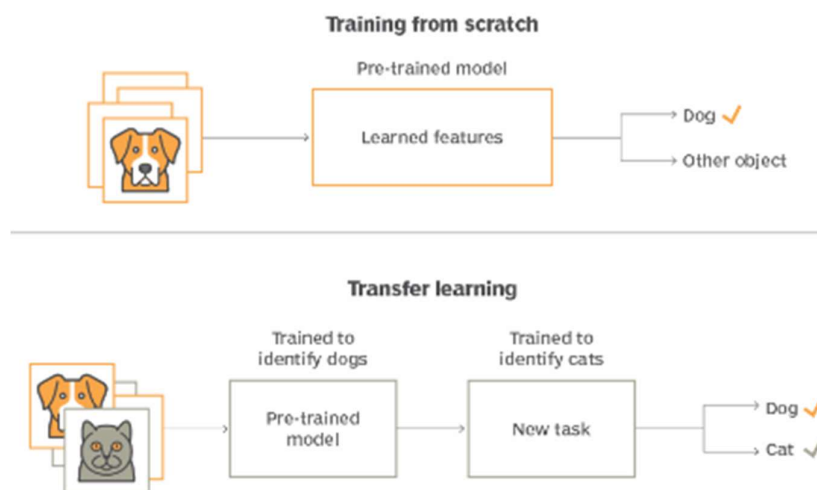


Figure-5: Transfer learning

Capacity to solve the problem of data scarcity is one of its main advantages. Large volumes of labelled data are usually needed for deep learning model training from start, but they may not always be accessible, particularly for specific tasks or domains. Transfer learning enables practitioners to take advantage of the abundance of knowledge captured by the pre-trained model, even in situations where only a small quantity of task-specific data is available. This is achieved by beginning with a pre-trained model that has been trained on a large, general-purpose dataset (such as ImageNet for image classification tasks). As a result, training an efficient model takes a lot less time and money.

Moreover, transfer learning may result in quicker convergence and better generalization. The model starts with a head start, already knowing about low-level characteristics and patterns that are probably relevant to the current work, thanks to the initialization of parameters learnt from a previous task. As a result, the model may more quickly adjust to the subtleties of the intended job and perform better after fewer training cycles.

In computer vision problems, transfer learning is especially common since pre-trained convolutional neural networks (CNNs) like VGG, ResNet, and Inception are frequently utilized as feature extractors. To fine-tune the models on fresh datasets for tasks like object identification, segmentation, and picture classification, practitioners can remove the fully connected layers at the top of these networks and replace them with task-specific layers.

It has been effectively used in speech recognition, natural language processing (NLP), and image-related tasks, among other areas. For instance, in NLP, pre-trained language models like BERT and GPT have been refined for certain downstream tasks like text categorization, question answering, and sentiment analysis. This has allowed for the production of state-of-the-art results with a minimal amount of data and computational resources.

It has some benefits, but it also has drawbacks. Domain shift is a prevalent difficulty whereby the data distributions of the source and destination domains diverge. Under such circumstances, the representations of the pre-trained model could not transfer effectively to the target domain, resulting in less-than-ideal performance. By matching the source and target domains, domain adaptation approaches like adversarial training or data augmentation can help lessen this problem.

All things considered, transfer learning provides a useful and effective method for creating high-performance machine learning models, particularly in situations when obtaining labelled data is expensive or difficult. Transfer learning allows practitioners to obtain state-of-the-art outcomes across a wide variety of applications and domains, increase generalization, and speed model development by utilizing pre-trained models and transferring information between tasks.

5.3 Pre-Trained CNN

With their many designs and functionalities, pre-trained Convolutional Neural Networks (CNNs) have become highly effective instruments for computer vision applications. These include AlexNet, GoogLeNet, VGG16, and Inception-v3, each of which is built to be very proficient in a particular area of image identification and classification.

In AlexNet, three fully connected layers with Rectified Linear Unit (ReLU) activation functions and five convolutional layers with max pooling comprise the groundbreaking CNN architecture known as AlexNet. It is noteworthy that it includes 3x3, 5x5, and 11x11 convolutional filters, which allow features to be extracted at different sizes. This varied filter size approach improves the network's capacity to recognize complex features and patterns in pictures.

A new inception module called GoogLeNet consists of many concurrent convolutional layers with different filter sizes, pooling, and concatenation. This approach encourages resilience and adaptability in picture recognition tasks by making it easier to learn features at various resolutions and sizes concurrently. In order to prevent overfitting and enhance generalization performance, GoogLeNet also adds auxiliary classifiers to its intermediary layers. This allows the network to collect more discriminative features.

The 16-layer CNN architecture known as VGG16 has become well-known because to its efficiency and ease of use. VGG16, which was pre-trained on the ImageNet database, has an extensive comprehension of a variety of visual ideas, ranging from commonplace items to animals. Its capacity to categorize photos into a thousand distinct groups demonstrates the range of its feature representations and provides in-depth information about the content of photographs. Additionally, VGG16 can handle input picture sizes up to 224×224 pixels, which makes it appropriate for a range of datasets and applications.

A significant development in CNN architecture, Inception-v3 is distinguished by its complex and detailed design. With its 316 layers, Inception-v3 can extract subtle characteristics and patterns from input photos and categorize them into 1000 categories. It was trained on a subset of the ImageNet collection. After undergoing intensive training on more than a million photos, Inception-v3 has proven to be adaptable and successful in a variety of scenarios by displaying a high degree of competency in image recognition tasks.

All things considered, pre-trained CNNs like AlexNet, GoogLeNet, VGG16, and Inception-v3 have transformed computer vision by providing academics and practitioners with strong instruments for interpreting, classifying, and recognizing images. Every architecture contributes unique qualities and innovations that further machine learning and open up new avenues for research and discoveries in the field of visual intelligence.

6 Datasets

6.1 Facial Image Datasets

The datasets used in this study were picked according to certain standards meant to fully capture real-world situations. A large number of classes, a range of lighting settings, a range of face expressions, and a combination of quantitative and qualitative data are some of these requirements. The datasets strive to replicate the intricacies seen in real-world scenarios by integrating these characteristics and conditions, guaranteeing the resilience and suitability of the models created.

The Extended Yale Face Database B (Cropped) and the Extended Cohn-Kanade Dataset (CK+) are the two main datasets used in this investigation. These datasets are made up of several picture sets that are separated into test and training sets. The labelled data that makes up the training set gives the models the known ground truth labels they need to be trained. On the other hand, the test set functions as a separate evaluation set that allows the models' performance and accuracy to be evaluated.

The models are subjected to a broad range of changes that are frequently encountered in real-world circumstances because to the dataset's inclusion of various lighting conditions and unique face expressions. By learning robust representations and patterns from this exposure, the models become more capable of generalizing to new data. Furthermore, the datasets are enhanced by the addition of both quantitative and qualitative data, enabling a thorough evaluation of the models' performance using a variety of metrics and criteria.

All things considered, the deliberate selection of datasets according to certain attributes and situations seeks to offer a realistic and demanding environment for model building and assessment. These datasets allow researchers to create and evaluate models that can handle the subtleties and complexity seen in real-world applications of face recognition and related tasks by simulating problems and scenarios.

6.1.1 *The Extended Yale Face Database B(Cropped):*

There are many lighting settings in the Extended Yale Face Database B (Cropped), which represent actual lighting situations. Pre-processing is done on each photograph with great care, involving manual alignment, cropping, and scaling to bring the facial contour standard to 168x192 pixels. This guarantees consistency and homogeneity throughout the dataset, which makes analysis and model training easier.

After pre-processing, damaged photographs are found and removed from the dataset, leaving a final set of 2404 pictures that represent 38 distinct people. The dataset is then divided into the training set and the test set, two separate groups. Eighty percent of the photographs, or 1910 photos, make up the training set, while the remaining twenty percent, or 494 images, make up the test set.

To provide a distinct separation between the data used for model training and the data used for assessment, this segmentation was justified. The training set provides plenty of instances for learning and parameter optimization, laying the groundwork for the growth of the model. On the other hand, the test set acts as a separate validation set that makes it possible to evaluate the model's performance on hypothetical data. Accurately assessing the robustness and generalization capacities of the trained models depends on this distinction.

The Extended Yale Face Database B (Cropped), with its meticulous dataset segmentation and careful inclusion of a variety of lighting conditions, provides a thorough and demanding environment for face recognition research. Standardized pre-processing procedures and careful data curation

enhance the dataset's quality and dependability, enabling significant discoveries and developments in the domains of computer vision and facial recognition technologies.

6.1.2 *The Extended Cohn-Kanade Dataset (CK+):*

A library of face images called the Extended Cohn-Kanade Dataset (CK+) was created with the express purpose of capturing a broad spectrum of human emotions. CK+ was created in 2000 as an addition to the Cohn-Kanade (CK) database. It consists of 593 sequences with 123 individuals, all of whom display unique emotional expressions. After careful emotional analysis, these visual sequences are divided into six main categories: disgust, fear, happiness, sadness, and surprise.

Even though there are six emotions included in the collection, not every contributor provides an image for every mood. The emphasis of the dataset is reduced to the top four emotions that the majority of participants stated in order to simplify it. Only those who exhibit all four of the chosen emotions are kept for further examination. Happy (2043 photographs from 107 individuals), surprised (1946 photos from 115 individuals), fear (1716 photos from 94 individuals), and disgust (1710 photos from 92 individuals) are some examples of these feelings. Interestingly, 18 people provide pictures that show each of the four face emotions.

Three distinct phrases are used for training, which guarantees a varied representation of facial emotions across the dataset. One expression is left out during testing in order to assess model performance thoroughly. As a result, 4205 images representing a range of emotional states make up the training set, which offers a rich and varied dataset for developing and accessing facial recognition algorithms.

The careful selection of CK+ guarantees a balanced depiction of participants and emotions, promoting a thorough comprehension of facial expressions in people. Through its emphasis on the most common emotions and integration of various training and testing methodologies, CK+ provides researchers with an invaluable tool for progressing the domain of emotion identification and comprehending human conduct via facial expressions.

7 Methodology

In this section the main model architecture discussed, the Person recognition system using Deep learning algorithms. The Experiment was split into two approaches.

7.1 *MATLAB Code:*

This MATLAB script uses the YOLOv4 object detector to recognize people in each frame, making it easier to collect images from a camera in real time.

```
clc;
clear;

datasetFolder = 'dataset';
if ~exist(datasetFolder, 'dir')
    mkdir(datasetFolder);
end
try
    % Webcam
    cam = webcam(1);
    preview(cam);
```

```

% Counter for limiting the number of frames
frameCount = 0;
maxFrames = 200;
yolov4Detector = yolov4ObjectDetector("tiny-yolov4-coco");
% Initialize variables for dataset
imageIndex = 0;
% Track multiple people and save images for the dataset.
while true
    frame = snapshot(cam); % Capture frame
from webcam
    bboxes = helperDetectObjects(yolov4Detector, frame); % Detect people
    detectedFrame = insertObjectAnnotation(frame, 'rectangle', bboxes,
'Person', 'LineWidth', 3); % Display bounding boxes
    imshow(detectedFrame);
    for i = 1:size(bboxes, 1)
% Save images with bounding boxes containing people
        personBox = bboxes(i, :);
% Extract each person's bounding box
        personImage = imcrop(frame, personBox);
% Crop the person from the frame
        imageFilename = fullfile(datasetFolder, ['person_' num2str(imageIndex)
'.png']); % Save the cropped image to the dataset folder
        imwrite(personImage, imageFilename);
        imageIndex = imageIndex + 1;
% Increment image index
    end
end
catch ME
    % Error handling
    disp('An error occurred:');
    disp(ME.message);
end
% Release webcam
clear cam;
function box = helperDetectObjects(yolov4Det, frame)
    [box, ~, class] = detect(yolov4Det, frame, 'Threshold', 0.5, 'MinSize', [5 5]);
    box = box(class == "person", :);
end
function key = getkeywait()
    w = waitforbuttonpress;
    key = char(w);
end

```

- Initialization: The script creates a directory called "dataset" to hold the stored photographs and clears the workspace to begin.
- camera Setup: After the camera has been initialized, live preview capabilities is enabled so that you may watch the frames that have been taken.
- Object Detection: The script recognizes people in every camera frame by using the YOLOv4 object detector. To ensure accurate person detection, it uses a confidence criterion of 0.5 to weed out detections that are lower than this threshold. To further improve the detecting process, a minimum bounding box size of 5x5 pixels is chosen.

- **Image Extraction:** The script crops the relevant regions of interest from the camera frames when it detects people. All of these resized photos of each individual are then stored as PNG files in the specified 'dataset' folder. An index is assigned to each stored image, which starts at 0 and increases with each picture that is taken.
- **Error Management:** To ensure seamless functioning, error handling techniques are built into the script to intercept and show any observed faults during execution.
- **Cleanup:** To ensure appropriate resource management, the script releases the webcam resource and removes the related camera object from the workspace as a last step.

The script includes two auxiliary functions in addition to its primary functions:

- **helperDetectObjects:** This function filters detections identified as "person" and extracts the appropriate bounding boxes for additional processing after processing the output from the YOLOv4 detector.
- **getkeywait:** Designed to wait for button press events, this function records the pushed key, if any, to enable user interaction.

All things considered, this script provides a simple yet reliable way to record and store photos of recognized people in live webcam frames. Because it uses YOLOv4 to assure reliable person detection, it may be used for a variety of computer vision applications, such as creating training and assessment datasets.

```
% Load pre-trained VGG-16 model
net = vgg16;

% Load and preprocess the dataset (assuming 'dataset' folder contains images of
people)
imds = imageDatastore('dataset', 'IncludeSubfolders', true, 'LabelSource',
'foldernames');
imds.ReadFcn = @(filename)imresize(imread(filename),
net.Layers(1).InputSize(1:2));

% Split the dataset into training and validation sets (80% training, 20%
validation)
[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.8, 'randomized');

% Replace the fully connected layer of the VGG-16 network with a new fully
connected layer
numClasses = numel(categories(imdsTrain.Labels));
layers = [
    net.Layers(1:end-3)
    fullyConnectedLayer(numClasses, 'Name', 'fc8', 'WeightLearnRateFactor', 10,
'BiasLearnRateFactor', 10)
    softmaxLayer('Name', 'softmax')
    classificationLayer('Name', 'classoutput')
];

% Set training options
options = trainingOptions('sgdm', ...
    'MiniBatchSize', 32, ...
    'MaxEpochs', 10, ...
```

```

'InitialLearnRate', 1e-4, ...
'ValidationData', imdsValidation, ...
'ValidationFrequency', 10, ...
'ValidationPatience', Inf, ...
'Verbose', true, ...
'Plots', 'training-progress');

% Train the network
netTransfer = trainNetwork(imdsTrain, layers, options);

% Classify images in the validation set and calculate accuracy
YPred = classify(netTransfer, imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = mean(YPred == YValidation);
disp(['Validation Accuracy: ' num2str(accuracy*100) '%']);

% Perform person recognition on new images
newImage = imread('F:\RAJESH RAJ\RP\Person Detection using deep
learning\dataset\new_image.jpg');
resizedImage = imresize(newImage, net.Layers(1).InputSize(1:2));
prediction = classify(netTransfer, resizedImage);
disp(['Predicted Label: ' char(prediction)]);

```

Code snippet showing how to use a pre-trained VGG-16 model and an accessible dataset to conduct person recognition using transfer learning.

Using the supplied dataset, this code refines a pre-trained VGG-16 model, making use of transfer learning. The dataset is divided into training and validation sets, a new fully connected layer appropriate for the number of classes in the dataset is substituted for the fully connected layer of VGG-16, and stochastic gradient descent is used to train the network. Lastly, it applies person recognition to a fresh image and assesses the model's accuracy on the validation set.

- 7.2 **Approch-1:** multi-layer perceptron neural networks (MLP) were used, and each dataset was split into training and test sets. Model training was conducted using the training set.
- 7.3 **Approch-2:** Applying deep neural networks for feature extraction transfer: CNN+MLP: This method incorporated the feature extraction process utilizing pre-trained CNN (Inception-V3) to the same dataset as III-B1. The pre-trained CNN was used to extract the training set's characteristics. Images are flattened into a feature vector by the last layer of a pre-trained CNN. The classification model was then trained using each and every training feature vector. Lastly, the test set, which was also retrieved in the same manner, was used to assess the model.

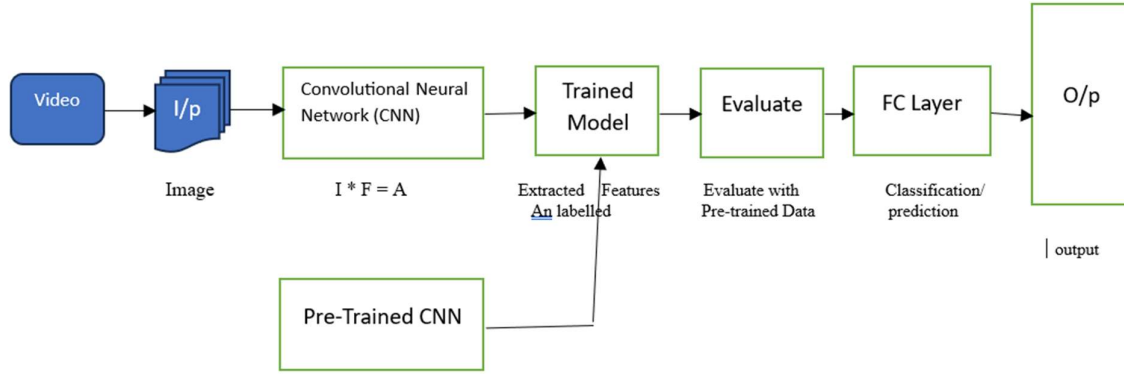


Figure-6: CNN+Transfer learning (Project model)

- 7.4 **Learning rate:** In a losing trend, the learning rate is a continuous measurement. When the learning rate exceeds the optimal value, it indicates that the overshoot is preventing it from finding the least loss. It will take a while to reach the minimum, though, if it is less than the appropriate value. This experiment shows the optimal learning rate value that works with the categorization model. The parameter splits to five values: 0.00001, 0.0001, 0.001, 0.01, and 0.1. It is used to test the loss function.

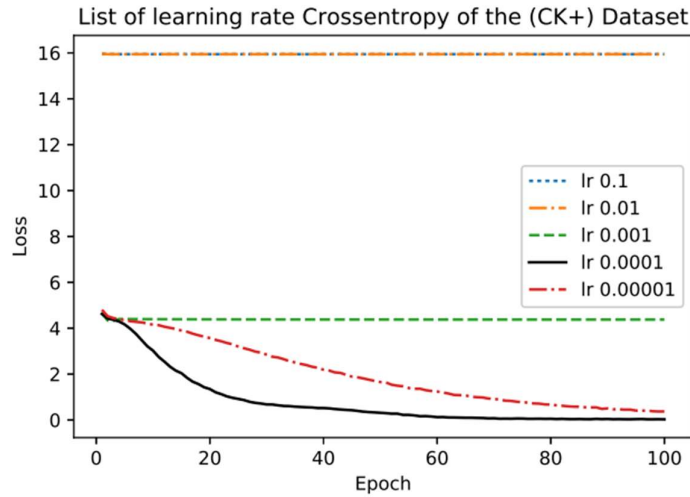


Figure-7: Dropout cross-entropy of (CK+) dataset

The result of cross entropy of different Learning rate values,

Learning Rate	Training Loss	Testing Loss
0.1	15.9418	15.9033
0.01	15.9418	15.9033
0.001	4.2163	4.1334
0.0001	0.0313	0.0048
0.00001	0.3756	0.0552

Table-1

Table-I displays the test dataset's loss, or cross-entropy, for various learning rate values. Following a few epochs of model training, the loss value of the test dataset dramatically decreased, leading us to conclude that the suited learning rate is equal to 0.0001.

7.5 **Dropout:** The overfitting issue can be avoided by using the dropout technique. Figure that follows displays the accuracy outcomes for various dropout settings. The learning rate in this experiment was fixed at 0.0001. During training, we can observe that the network does not appear to overfit to the training dataset. High accuracy results may be obtained with a modest dropout value.

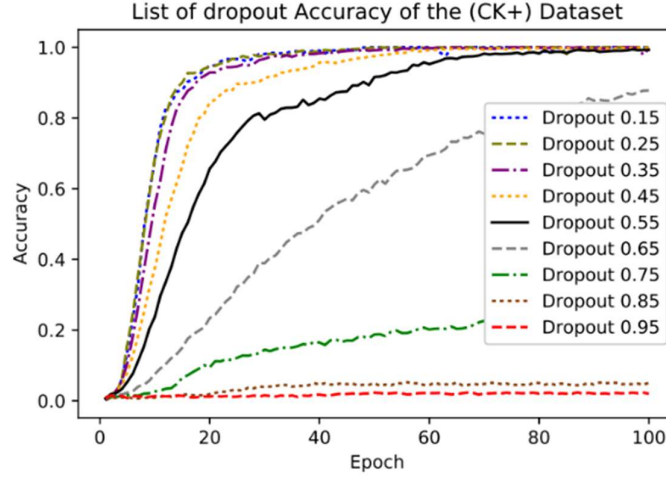


Figure-8: Drop accuracy

The accuracy results utilizing various dropout settings are displayed in Table II. Numerous applicable dropout values with good accuracy were discovered, including 0.15, 0.25, 0.35, 0.45, 0.55, and 0.65.

On the other hand, the values of 0.75, 0.85, and 0.95 fall into the category of underfitting. As a result, we represent the dropout value in the trials by averaging those appropriate values, which equal 0.4.

Dropout	Training Accuracy	Testing Accuracy
0.15	1.0000	0.9993
0.25	0.9998	0.9986
0.35	0.9983	0.9972
0.45	0.9969	1.0000
0.55	0.9941	0.9909
0.65	0.8930	1.0000
0.75	0.2373	0.4804
0.85	0.0490	0.0968
0.95	0.0233	0.0105

Table-II

In the experiments, the multilayer perceptron (MLP) structure was made up of three hidden layers, each containing 1024, 512, and 256 nodes. Rectified Linear Units (ReLU) serve as each node in

the buried layer's activation function. The number of samples that were propagated through the network in each step was 16, and this value was specified as the batch size. The machine learning algorithm was a gradient descent method with a learning rate of 0.0001. Additionally, we employed the dropout approach, which is the set to 0.4 for each hidden layer in neural networks, to lessen the overfitting issue.

8 Work Results

The advantages of applying the transfer learning in a large number of classes with a limited sampled size per class were the main topic of this research. The outcomes of employing several neural network models were influenced by a number of factors, including feature spaces, loss surfaces, and starting weights. As a result, every experiment made maximum use of the same environment by exchanging parameters.

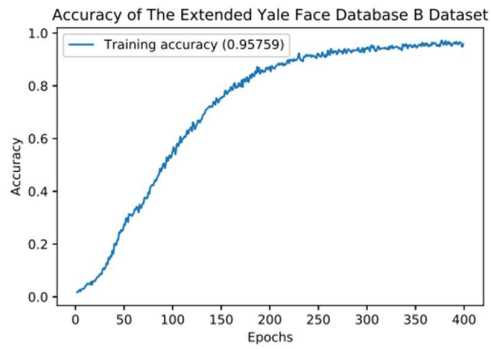


Figure-9: Accuracy on training set of Extended Yale Face Database-B

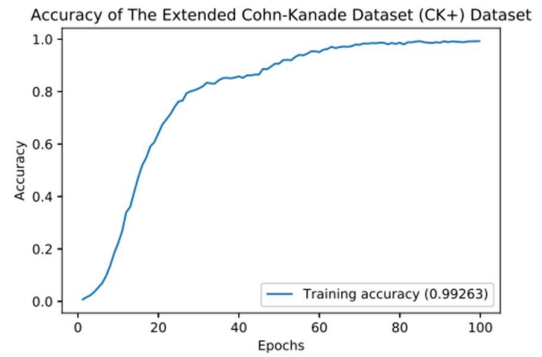


Figure-10: Accuracy on training set of Extended Cohn-Kanade Dataset (CK+)

8.1 The cross entropy results

The model's loss function is denoted by the Cross-Entropy. As indicated in Table III, Cross-Entropy of the test set utilizing the model in method 2 was much lower than that of the test set utilizing the model in approach 1. As seen in Figures 7 and 8, respectively, the losses value was somewhat decreased after training over 200 and 60 epochs for the Yale-B and CK+ datasets.

8.2 Accuracy result

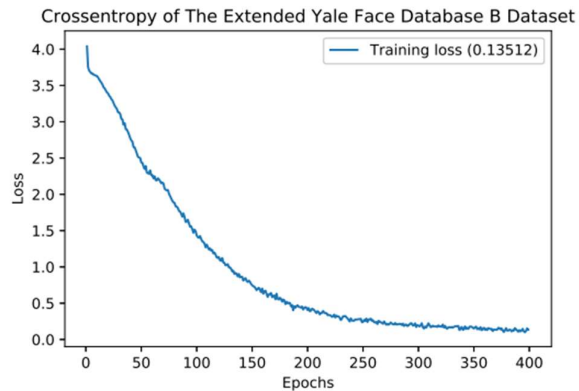


Figure-11: Loss values on the training set of Extended Yale Face Database-B

On both datasets, the accuracy of the test set using the model in method 2 was likewise much higher than accuracy of the test set using the model in approach 1. Figure 9 illustrates that after training the model in method 2 over 200 epochs, the accuracy values of the training set for the Yale-B dataset were greater than 80%. Just 30 epochs are required for the CK+ dataset with this approach. On both datasets, the accuracy of the test set and the training set are equal. Given that the model can accurately predict the unknown data, it is likely not overfit the training set.

Dataset	Model	Training Set		Testing Set		Computational Times (s)
		Cross-Entropy	Accuracy (%)	Cross-Entropy	Accuracy (%)	
Yale	MLP	15.63	3.00	15.68	2.70	15870 ^c
	CNN + MLP	0.135	95.76	0.126	96.56	957 ^a +251 ^b +3770 ^c = 4978
CK+	MLP	15.98	0.88	15.79	2.05	28709 ^c
	CNN + MLP	0.022	99.27	0.001	99.93	2305 ^a +784 ^b +2581 ^c = 5670

Table-3: Test and Training comparison

8.3 The computational time results

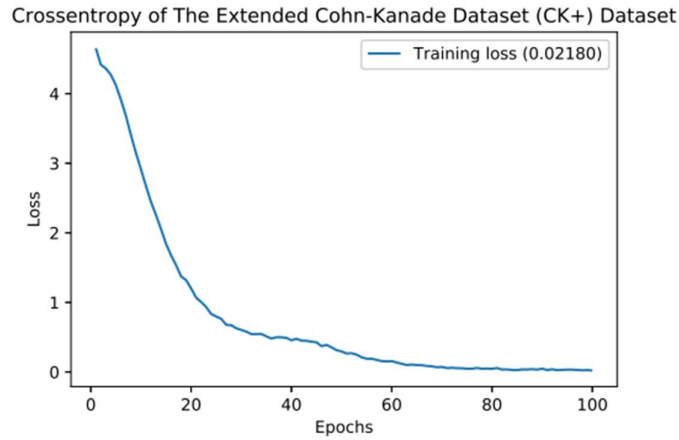


Figure-12: Result for model loss on the training of Extended Cohn-Kanade Dataset (CK+)

The model training time is the sole value that represents the whole computing time of approach 1. Approach 2's entire calculation time is represented by three values: 1) The processing time for extracting features from the training dataset; 2) The processing time for extracting features from the testing dataset, and 3) The processing time for training the model. The overall time values in method 2 are lower than the one in approach 1, despite the fact that models in approach 2 include numerous time values.

9 Conclusion

Person recognition has been studied utilizing transfer learning with a pre-trained convolution neural network as the feature extractor. Two widely recognized public face datasets were utilized to assess how well the conventional and transferred learning models performed. In the face recognition

issue, there are numerous classes, but each class only has a limited sample size. It may infer from the trials that face recognition using transferred deep feature learning produces excellent results. Compared to the classic approach, the transferred learning model performs significantly better. Furthermore, there is no indication that either dataset has an overfitting issue.

10 Acknowledgement

My profound appreciation is given to Prof. Dipti Patra of the National Institute of Technology, Rourkela, for her great advice and steadfast assistance during the creation of the Person Recognition System employing CNN and Transfer Learning. Her knowledge and guidance have been invaluable in helping to shape our project. Her sage advice and encouragement have also been tremendously appreciated, and they have improved my learning process. Without her direction, this project would not have been feasible, and I sincerely appreciate the chance to work under her direction.

11 Future Scope

In future works, pre-trained models and improve the current one in my next work. Moreover, non-facial components of the original CK+ dataset comprise the ears, head or cap, a portion of the neck, and a portion of the backdrop picture. But these features are unnecessary for categorization and have to be eliminated; just the area with the mouth, nose, and eyes should be utilized.

12 Reference

1. West, Jeremy; Ventura, Dan; Warnick, Sean (2007). "[Spring Research Presentation: A Theoretical Foundation for Inductive Transfer](#)". Brigham Young University, College of Physical and Mathematical Sciences. Archived from [the original](#) on 2007-08-01. Retrieved 2007-08-05.
2. Russakovsky, O., Deng, J., Su, H., et al. "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision (IJCV)*. Vol 115, Issue 3, 2015, pp. 211–252
3. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
4. *Very Deep Convolutional Networks for Large-Scale Visual Recognition* http://www.robots.ox.ac.uk/~vgg/research/very_deep/
5. V. Radhamani and G. Dalin, "A supporting survey to step into a novel approach for providing automated emotion recognition service in mobile phones," in 2018 2nd International Conference on Inventive Systems and Control (ICISC), Jan 2018, pp. 35–39.
6. T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of training convolutional neural networks," CoRR, vol. abs/1506.01195, 2015. [Online]. Available: <http://arxiv.org/abs/1506.01195>
7. M. Y. W. Teow, "Understanding convolutional neural networks using a minimal model for handwritten digit recognition," in 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS), Oct 2017, pp. 167–172.
8. J. R. C. P. de Oliveira and R. A. F. Romero, "Transfer learning based model for classification of cocoa pods," in 2018 International Joint Conference on Neural Networks (IJCNN), July 2018, pp. 1–6.

9. G. Gautam and S. Mukhopadhyay, "Contact lens detection using transfer learning with deep representations," in 2018 International Joint Conference on Neural Networks (IJCNN), July 2018, pp. 1–8.
10. G. Gautam and S. Mukhopadhyay, "Contact lens detection using transfer learning with deep representations," in 2018 International Joint Conference on Neural Networks (IJCNN), July 2018, pp. 1–8.
11. Q. Li, L. Mou, K. Jiang, Q. Liu, Y. Wang, and X. X. Zhu, "Hierarchical region based convolution neural network for multiscale object detection in remote sensing images," in IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, July 2018, pp. 4355–4358.
12. S. D., "Understanding convolutional neural networks," In Seminar Report, Informatik und Naturwissenschaften Lehr-und Forschungsgebiet Informatik VIII Computer Vision., 2014.
13. S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. on Knowl. and Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
14. [Online]. Available: L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 5, pp. 1019–1034, May 2015.
15. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.
16. Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, "Building high-level features using large scale unsupervised learning," in International Conference in Machine Learning, 2012.
17. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
18. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," CoRR, vol. abs/1409.4842, 2014. [Online]. Available: <http://arxiv.org/abs/1409.4842>.
19. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," CoRR, vol. abs/1502.03167, 2015. [Online]. Available: <http://arxiv.org/abs/1502.03167>.
20. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," CoRR, vol. abs/1512.00567, 2015. [Online]. Available: <http://arxiv.org/abs/1512.00567>
21. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, June 2010, pp. 94–101. F. Chollet et al., "Keras," <https://keras.io>, 2015
22. https://en.wikipedia.org/wiki/Facial_recognition_system#History_of_facial_recognition_technology
23. <https://iopscience.iop.org/article/10.1088/1742-6596/1755/1/012006/meta>

24. <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>
25. <https://www.ibm.com/topics/convolutional-neural-networks>