Faculty of Computer Science and Technology

# *Ethnicity Classification using Deep Learning*

**Christa Tracy Mojikon**

**Bachelor of Computer Science with Honours**

**(Computational Science)**

2025

**ETHNICITY CLASSIFICATION USING DEEP LEARNING**

**CHRISTA TRACY MOJIKON**

This project is submitted in partial fulfilment of the

requirements for the degree of

Bachelor of Computer Science and Information Technology

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

**UNIVERSITI MALAYSIA SARAWAK**

## THESIS STATUS ENDORSEMENT FORM

**TITLE** ETHNICITY CLASSIFICATION USING DEEP LEARNING

**ACADEMIC SESSION:** 2024/2025

CHRISTA TRACY MOJIKON

**(CAPITAL LETTERS)**

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [ or for the purpose of interlibrary loan between HLI ]
5. ** Please tick ( √ )

| | | |
|---|---|---|
| ☐ | CONFIDENTIAL | (Contains classified information bounded by the OFFICIAL SECRETS ACT 1972) |
| ☐ | RESTRICTED | (Contains restricted information as dictated by the body or organization where the research was conducted) |
| ✓ | UNRESTRICTED | |

Validated by

Christa

(AUTHOR'S SIGNATURE)

(SUPERVISOR'S SIGNATURE)

Permanent Address

Lorong 2, Taman Wirajaya, No 21,
89257, Tamparuli, Sabah,
Malaysia.

Date: 17 July 2025

Date: 17 July 2025

Note * Thesis refers to PhD, Master, and Bachelor Degree
** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

## TABLE OF CONTENTS

**LIST OF FIGURES**

## LIST OF TABLES

# ACKNOWLEDGMENT

I would like to begin by sincerely thanking Dr. Chiew Kang Leng, my project supervisor, for their unwavering support, valuable guidance, and constructive feedback throughout the duration of this study. Their knowledge and mentorship were instrumental to the completion of this project.

My heartfelt appreciation also goes to my examiner, Dr. Suhaila binti Saee, for dedicating their time to review and assess this work. I am truly grateful for their input and thoughtful evaluation.

I am also thankful to my friends and fellow students for their constant encouragement, collaborative spirit, and insightful discussions, all of which greatly contributed to the progress of this project.

Most importantly, I am deeply indebted to my family for their endless support, patience, and belief in my abilities. Their encouragement has been my greatest motivation throughout this journey.

## ABSTRACT

Classifying ethnicity is an important field of study with various practical applications in identity verification, personalized user experiences, demographic studies, and bias mitigation in artificial intelligence (AI) systems. As facial recognition technology becomes increasingly integrated into societal functions, ensuring accurate and fair ethnicity classification is essential to prevent the perpetuation of biases and inaccuracies. This study explores ethnicity classification using deep learning, utilizing a Convolutional Neural Network (CNN) built upon the VGG-Face architecture, which is widely recognized for its strong performance in facial recognition applications. Despite advancements in ethnicity classification, existing methods often lack diverse datasets and struggle with generalizability, limiting their effectiveness in multicultural societies like Malaysia. To address these gaps, this research aims to develop and evaluate a deep learning model capable of accurately classifying ethnicities in Malaysia, focusing on major ethnic groups such as Malay, Chinese, and Indian. A custom dataset representing Malaysia's ethnic diversity was curated to ensure inclusivity and fairness, overcoming the limitations of global datasets that lack representation of local ethnic characteristics. These findings highlight the capabilities of deep learning to improve demographic data collection and analysis. By enabling more equitable resource allocation and better-tailored services, this study adds value to the domain of demographic analysis by providing a scalable, accurate, and ethical solution for ethnicity classification. Its applications extend to governance, healthcare, and social services, contributing to efforts that promote inclusivity and improve service delivery in Malaysia’s diverse population.

# ABSTRAK

Klasifikasi etnik ialah bidang kajian yang penting dengan pelbagai aplikasi praktikal dalam pengesahan identiti, pengalaman pengguna yang diperibadikan, kajian demografi dan pengurangan bias dalam sistem kecerdasan buatan (AI). Dengan perkembangan teknologi pengecaman wajah yang semakin meluas dalam fungsi sosial, keperluan untuk memastikan klasifikasi etnik yang tepat dan adil adalah penting untuk mengelakkan ketidaktepatan dalam teknologi ini. Kajian ini meneroka klasifikasi etnik menggunakan kaedah pembelajaran mendalam, dengan menggunakan Rangkaian Neural Konvolusi (CNN) berasaskan seni bina VGG-Face, yang terkenal dengan prestasinya yang tinggi dalam tugas pengecaman wajah. Walaupun terdapat kemajuan dalam klasifikasi etnik, kaedah yang sedia ada sering kekurangan dataset yang pelbagai dan menghadapi masalah dalam generalisasi seterusnya mengehadkan keberkesanannya dalam masyarakat berbilang budaya seperti Malaysia. Untuk menangani jurang ini, penyelidikan ini bertujuan untuk membangunkan dan menilai model pembelajaran mendalam yang mampu mengklasifikasikan etnik di Malaysia dengan tepat, memfokuskan kepada kumpulan etnik utama seperti masyarakat Melayu, Cina, dan India. Dataset khas yang mewakili kepelbagaian etnik Malaysia telah dikumpulkan untuk memastikan keterangkuman dan keadilan bagi mengatasi had set data global yang tidak mempunyai perwakilan ciri etnik tempatan. Penemuan kajian ini menyerlahkan potensi pembelajaran mendalam untuk meningkatkan pengumpulan dan analisis data demografi. Dengan membolehkan pembahagian sumber yang lebih adil dan perkhidmatan yang disesuaikan, penyelidikan ini menyumbang kepada bidang analisis demografi dengan menawarkan penyelesaian berskala, tepat dan beretika untuk klasifikasi etnik. Aplikasinya merangkumi bidang pentadbiran, kesihatan dan perkhidmatan sosial, seterusnya menyokong dalam penambahbaikan penyampaian perkhidmatan dalam masyarakat berbilan kaum di Malaysia.

# CHAPTER 1

# INTRODUCTION

## 1.1 Background Study

Demographic data is fundamental to addressing societal needs and enabling informed decision-making across various sectors, including governance, healthcare, education, and social services. In this context, ethnic classification plays a crucial role in ensuring resources are effectively tailored to specific communities, fostering inclusivity, and understanding societal composition. Ethnicity classification has the potential to improve demographic studies and assess the fairness of current methods used (Jain et al., 2022).

Advancements in machine learning and deep learning techniques, particularly through Convolutional Neural Networks (CNNs), have demonstrated significant potential in solving complex classification tasks based on images, including facial recognition and ethnicity prediction (Vo et al., 2018). The model excels in analyzing facial features to predict an individual's ethnicity, offering a foundation for automated solutions (Alzubaidi et al., 2021). However, existing technologies do not adequately address Malaysia's unique demographic diversity.

The challenges remain, particularly regarding dataset limitations, model accuracy, and ethical considerations surrounding privacy and bias (Wehrli et al., 2021). Existing models are often trained on generalized global datasets, which fail to capture Malaysia's distinct ethnic groups. This lack of representation results in reduced model effectiveness, poor generalization, and inaccuracies in predictions.

Moreover, manual identification methods, while suitable for small datasets, become impractical, time-intensive, and error-prone as the dataset grows (Li et al., 2024). These traditional approaches are also susceptible to human biases, fatigue, and cultural assumptions, further compromising their reliability for large-scale applications (Mehrabi et al., 2021).

The challenges underscore the need for automated, standardized methods to improve accuracy and scalability in ethnicity classification. Therefore, this project proposes an ethnicity classification system using a deep learning model specifically designed for Malaysia's diverse ethnic groups. A new dataset will be curated to represent the unique ethnic diversity of the region, emphasizing inclusivity and fairness with a more accurate and ethical solution for ethnic classification. The proposed solution aims to develop accurate ethnicity classification system to minimize human bias and provide scalable results for large datasets. In doing so, it will provide valuable insights to the fields of demographic research and machine learning.

### 1.2 Problem Statement

Demographic data, especially ethnicity, is essential for demographic analysis, diversity efforts and personalized services. However, in Malaysia, there is a significant gap in automated systems for classifying ethnic groups based on facial features. This gap limits the efficiency of identifying eligible recipients for diversity and inclusion programs. Current classification models rely on datasets that lack the representation of Malaysia's ethnic diversity, leading to poor generalization and inaccuracies (Sulfii et al., 2022). Datasets specifically focused on Malaysian ethnic diversity should be developed and used for training more accurate classification models.

While manual identification can work for small-scale tasks, it becomes highly impractical and time-consuming when dealing with large datasets. In ethnicity classification, it is essential to verify each record to ensure that it accurately represents an individual's characteristics. Reviewing each record manually will be an overwhelming process, especially when the dataset grows, and it can be influenced by personal biases, fatigue, or cultural assumptions. In contrast to traditional manual approaches, automated methods provide significant benefits in terms of faster processing speeds and the capacity to manage exceptionally large datasets (Chandrasekar et al., 2024). This will offer a more effective solution by ensuring consistent and objective classification. This reduces the potential for human error and bias, significantly improving the accuracy and reliability of the data for demographic analyses.

**1.3 Research Questions**

1. How does the choice of deep learning framework or algorithm impact the model cability to classify ethnic groups accurately?
2. Is there a significant relationship between the size and diversity of a training dataset and the accuracy of ethnicity classification for Malaysian ethnic groups?
3. How accurately can a deep learning model classify individuals into specific Malaysian ethnic groups based on facial features?

**1.4 Objectives**

1. Develop a deep learning model to classify different ethnic groups.
2. Construct a diverse dataset for training and testing the model.
3. Evaluate the model’s accuracy for better recognition of diverse ethnic backgrounds.

**1.5 Brief Methodology**

Development of ethnicity classification system using deep learning is structured into three main phases: data collection, model development, and model validation.

The first phase focuses on collecting and preparing dataset. Images will be gathered from various sources such as online sources, including search engines and social media platforms.

There will also be contributions from individuals who will voluntarily share their facial images along with their self-identify ethnicity for accurate labelling. Preprocessing steps will involve dataset cleaning, resize images, and normalization to standardize the input data (Chaki & Dey, 2018). Data augmentation techniques will be applied to enhance dataset diversity and improve the model's robustness.

The next phase will involve creating and training CNN model. The aim is to achieve satisfactory accuracy by fine-tuning settings like the learning rate and batch size. Figure 1.1 presents the flowchart of the proposed system. The system begins by retrieving face images from the dataset collected, which are then trained and processed to extract facial features, enabling the classification of new face images into distinct ethnic groups (Sulfii et al., 2022).

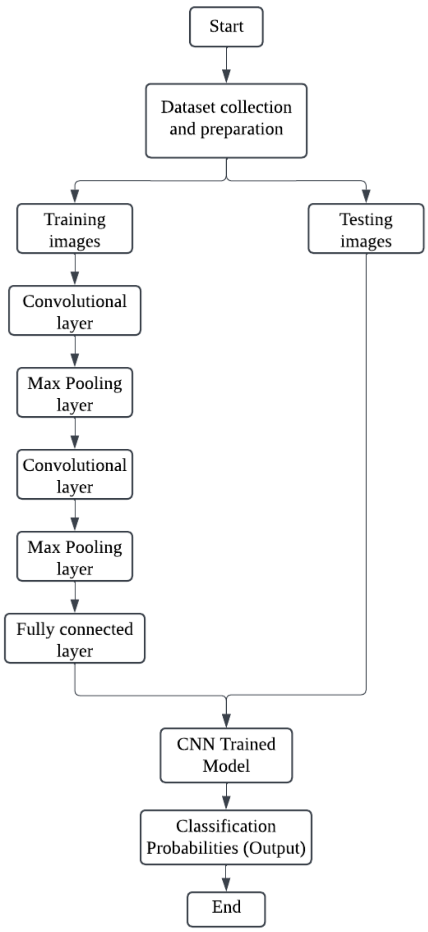



Figure 1.1 : Image Classification Flowchart (adapter from Sulfii et al., 2022)

The model consists of convolutional layers integrated with ReLU activation functions to effectively extract key features from the input images, followed by max pooling layers to reduce spatial dimensions. A fully connected layer is then added to correspond with the number of classification categories. The final output layer uses a softmax activation function to transform the model's output into probability scores, allowing the identification of the most probable class. The structure of the model is illustrated in Figure 1.2.

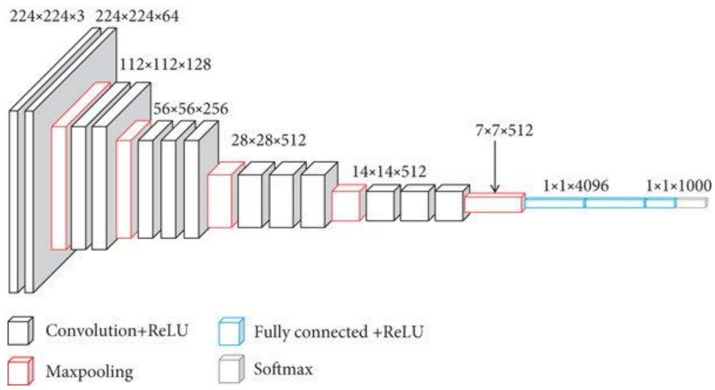



Figure 1.2: CNN Architecture with VGG Model (adapted from Mu et al., 2021)

Finally, the performance of the trained model will be assessed using an independent test dataset, focusing on metrics such as accuracy, precision, recall, and overall effectiveness (Makolo & Dada, 2023). Additionally, the model's outputs will be analyzed for potential biases to make sure the prediction across all ethnic groups is fair and accurate. Any identified biases will be addressed through further adjustments, such as rebalancing the dataset or refining the model architecture. Figure 1.3 outlines the different phases used in the methodology of the ethnicity classification using deep learning.

| PHASE | METHOD | OUTCOME |
|---|---|---|
| I. DATA COLLECTION | Collect facial images of different ethnicity from online sources. Apply the cleaning, resizing, and normalizing of the data collected. | A diverse and standardized dataset ready to be used for training and testing the model. |
| II. MODEL DEVELOPMENT | Develop and train a CNN with VGG architecture for facial images and ethnicity classification. | Trained deep learning model capable of extracting features of different ethnicities. |
| III. MODEL VALIDATION | Evaluate the model to measure its accuracy and precision. Refine the model based on test result. | A validated model with optimized performance and accurate ethnic classification. |

Figure 1.3: Brief Methodology for Ethnicity Classification Using Deep Learning

## 1.6 Project Scope

This study is focused on the development of a deep learning model to classify a few key ethnic groups in Malaysia, focusing on the Malay, Chinese, and Indian.

Given the constraints of data availability, the project will focus on a limited dataset of facial images representing these groups. This project seeks to create a functional model that can accurately classify these groups based on facial features and become a reliable proof-of-concept that addresses the country's specific demographic needs.

Since this project classification and accuracy will be for ethnic groups prominent within Malaysia, thus it may not generalize well to broader Asian or international populations. Additionally, the system's performance may vary based on diverse ethnic characteristics and environmental factors, impacting the accuracy of classifications in real-world applications.

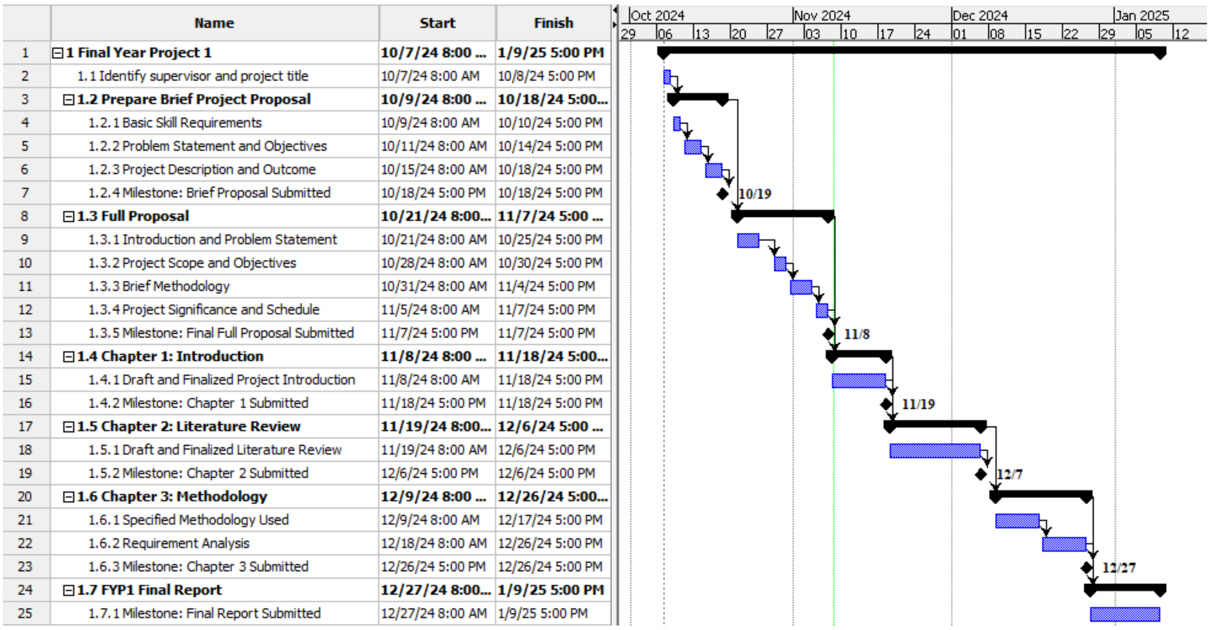
## 1.7 Significance of Project

This study is important due to its potential to develop an automated ethnicity classification using deep learning, providing an improved and precise approach for demographic data collection. By accurately identifying recipients across various ethnic groups, the model will improve resource allocation, ensuring that support is tailored to the specific needs of each community. This project aims to enhance ethnic group identification to improve resource allocation and support for diverse communities.

Furthermore, the deep learning techniques used in this system will be applicable to similar applications or problems in other areas or contexts beyond the original scope. This project also promotes a greater understanding of Malaysia's diverse ethnic landscape, highlighting the need for accurate representation and recognition of all groups. By improving upon the limitations of current systems, this project aims to develop a fair and unbiased technologies.

## 1.8 Project Schedule

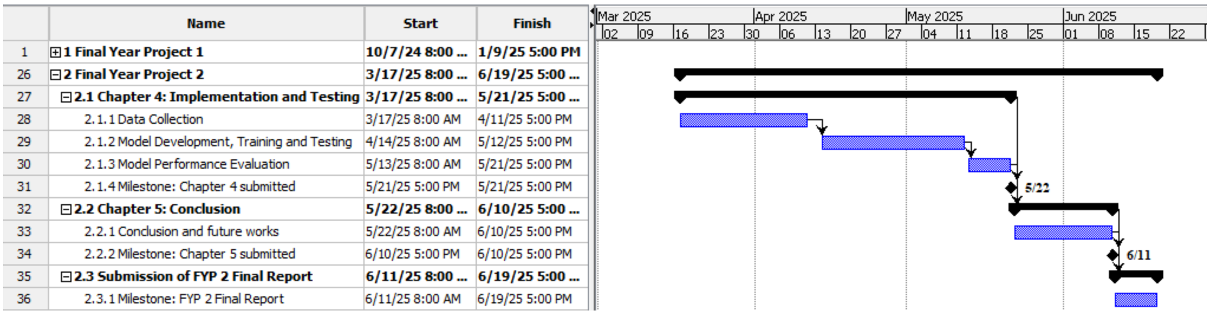
The Final Year Project (FYP) is structured into two primary phases: FYP 1 and FYP 2. In the first phase, the focus is on completing Chapter 1 until Chapter 3 of the FYP report writing. This involves the project introduction, literature review and project methodology. Figure 1.4 and Figure 1.5 below show the schedule for this project.



| | Name | Start | Finish |
|---|---|---|---|
| 1 | **1 Final Year Project 1** | **10/7/24 8:00 ...** | **1/9/25 5:00 PM** |
| 2 | 1.1 Identify supervisor and project title | 10/7/24 8:00 AM | 10/8/24 5:00 PM |
| 3 | **1.2 Prepare Brief Project Proposal** | **10/9/24 8:00 ...** | **10/18/24 5:00...** |
| 4 | 1.2.1 Basic Skill Requirements | 10/9/24 8:00 AM | 10/10/24 5:00 PM |
| 5 | 1.2.2 Problem Statement and Objectives | 10/11/24 8:00 AM | 10/14/24 5:00 PM |
| 6 | 1.2.3 Project Description and Outcome | 10/15/24 8:00 AM | 10/18/24 5:00 PM |
| 7 | 1.2.4 Milestone: Brief Proposal Submitted | 10/18/24 5:00 PM | 10/18/24 5:00 PM |
| 8 | **1.3 Full Proposal** | **10/21/24 8:00...** | **11/7/24 5:00 ...** |
| 9 | 1.3.1 Introduction and Problem Statement | 10/21/24 8:00 AM | 10/25/24 5:00 PM |
| 10 | 1.3.2 Project Scope and Objectives | 10/28/24 8:00 AM | 10/30/24 5:00 PM |
| 11 | 1.3.3 Brief Methodology | 10/31/24 8:00 AM | 11/4/24 5:00 PM |
| 12 | 1.3.4 Project Significance and Schedule | 11/5/24 8:00 AM | 11/7/24 5:00 PM |
| 13 | 1.3.5 Milestone: Final Full Proposal Submitted | 11/7/24 5:00 PM | 11/7/24 5:00 PM |
| 14 | **1.4 Chapter 1: Introduction** | **11/8/24 8:00 ...** | **11/18/24 5:00...** |
| 15 | 1.4.1 Draft and Finalized Project Introduction | 11/8/24 8:00 AM | 11/18/24 5:00 PM |
| 16 | 1.4.2 Milestone: Chapter 1 Submitted | 11/18/24 5:00 PM | 11/18/24 5:00 PM |
| 17 | **1.5 Chapter 2: Literature Review** | **11/19/24 8:00...** | **12/6/24 5:00 ...** |
| 18 | 1.5.1 Draft and Finalized Literature Review | 11/19/24 8:00 AM | 12/6/24 5:00 PM |
| 19 | 1.5.2 Milestone: Chapter 2 Submitted | 12/6/24 5:00 PM | 12/6/24 5:00 PM |
| 20 | **1.6 Chapter 3: Methodology** | **12/9/24 8:00 ...** | **12/26/24 5:00...** |
| 21 | 1.6.1 Specified Methodology Used | 12/9/24 8:00 AM | 12/17/24 5:00 PM |
| 22 | 1.6.2 Requirement Analysis | 12/18/24 8:00 AM | 12/26/24 5:00 PM |
| 23 | 1.6.3 Milestone: Chapter 3 Submitted | 12/26/24 5:00 PM | 12/26/24 5:00 PM |
| 24 | **1.7 FYP1 Final Report** | **12/27/24 8:00...** | **1/9/25 5:00 PM** |
| 25 | 1.7.1 Milestone: Final Report Submitted | 12/27/24 8:00 AM | 1/9/25 5:00 PM |



Figure 1.4: Project Schedule for FYP 1



| | Name | Start | Finish |
|---|---|---|---|
| 1 | ⊞1 Final Year Project 1 | 10/7/24 8:00 ... | 1/9/25 5:00 PM |
| 26 | ⊟2 Final Year Project 2 | 3/17/25 8:00 ... | 6/19/25 5:00 ... |
| 27 | ⊟2.1 Chapter 4: Implementation and Testing | 3/17/25 8:00 ... | 5/21/25 5:00 ... |
| 28 | 2.1.1 Data Collection | 3/17/25 8:00 AM | 4/11/25 5:00 PM |
| 29 | 2.1.2 Model Development, Training and Testing | 4/14/25 8:00 AM | 5/12/25 5:00 PM |
| 30 | 2.1.3 Model Performance Evaluation | 5/13/25 8:00 AM | 5/21/25 5:00 PM |
| 31 | 2.1.4 Milestone: Chapter 4 submitted | 5/21/25 5:00 PM | 5/21/25 5:00 PM |
| 32 | ⊟2.2 Chapter 5: Conclusion | 5/22/25 8:00 ... | 6/10/25 5:00 ... |
| 33 | 2.2.1 Conclusion and future works | 5/22/25 8:00 AM | 6/10/25 5:00 PM |
| 34 | 2.2.2 Milestone: Chapter 5 submitted | 6/10/25 5:00 PM | 6/10/25 5:00 PM |
| 35 | ⊟2.3 Submission of FYP 2 Final Report | 6/11/25 8:00 ... | 6/19/25 5:00 ... |
| 36 | 2.3.1 Milestone: FYP 2 Final Report | 6/11/25 8:00 AM | 6/19/25 5:00 PM |



Figure 1.5: Project Schedule for FYP 2

## 1.9 Project Outcome

The intended outcome of this project is a reliable and effective deep learning model that accurately classifies ethnic groups in Malaysia based on facial features. The model will streamline the process of ethnic classification, reducing the need for manual verification of demographic data and enabling organizations to collect accurate and reliable demographic data. This will improve resource allocation by ensuring that support is tailored to the specific needs of each ethnic group. Additionally, a diverse dataset reflecting Malaysia's ethnic diversity will be created with significant level of classification accuracy.

## 1.10 Project Outline

The first chapter introduces the project, covering its overview, problem statement, objectives, scope, significance, and anticipated outcomes. It sets the stage for the project by highlighting the importance of accurate ethnicity classification in promoting fairness and inclusivity, particularly in diverse societies like Malaysia.

The second chapter focuses on the literature review. It involves researching and analyzing existing ethnicity classification systems and comparing them to the proposed deep learning approach. This section examines how CNNs and other methodologies have been utilized in the past and identifies the gaps the proposed system aims to address.

The third chapter outlines the methodology adopted for the project, specifically detailing how deep learning approaches, such as convolutional neural networks, are used to classify ethnicities. It covers key development steps, including gathering and preparing data, training the model, and conducting evaluations, providing a clear workflow for creating an accurate and inclusive system.

Chapter four focuses on the model implementation of the ethnicity classification system using deep learning. It describes the use of VGG-Face for extracting facial features and the construction of a CNN model to categorize images into ethnic groups. Steps for training and deploying the system are also explained.

Chapter five focuses on the results obtained and their discussion concerning the ethnicity classification model. It includes an explanation of the testing procedures for model performance assessment. This section also analyzes the system's predictions for potential biases and describes the steps taken to mitigate them.

The final chapter concludes the project by summarizing its findings, evaluating whether the objectives were achieved. It also explores opportunities for future enhancements to handle more diverse populations and reduce biases further.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Background Study

Ethnicity classification is increasingly recognized as a significant research focus in the field of computer vision and machine learning, particularly in today's multicultural societies. This task involves identifying an individual's ethnic group based on facial features and gained significant attention due to its applications in demographic research, personalized services, and enhancing facial recognition systems. In countries like Malaysia, with its rich ethnic diversity, ethnicity classification holds relevance in fields such as national security, marketing, and healthcare, where demographic insights are vital.

Since the rise of deep learning, this domain has seen notable progress that enable models to perform complex classification with greater accuracy and efficiency. This literature review focuses on ethnicity classification using facial features and deep learning models to achieve accurate and reliable results. It examines the current state of the field, analyzing the characteristics, size, and diversity of datasets used in studies, and their impact on model performance. The review explores the relationships between datasets and models to mitigate bias and improve generalization. Additionally, it compares the performance of various deep learning architectures, training methodologies and metrics evaluation, providing insights into best practices and emerging techniques.

The aim of this review is to address key research questions regarding ethnicity classification using deep learning. Specifically, it seeks to identify the most suitable datasets for ethnicity classification, the most effective deep learning architectures and frameworks, and the application of transfer learning as well as data augmentation. The model performance would be assess using metrics evaluation to recognize any potential bias in ethnic classification models.

## 2.2 Deep Learning Framework

Deep learning frameworks offer essential tools for constructing, training, and evaluating deep neural networks, typically utilizing GPU-based libraries to achieve efficient training performance. TensorFlow, coupled with the Keras API and PyTorch are the common frameworks used for ethnicity classification tasks.

TensorFlow, when used with Keras, offers a highly user-friendly environment for developing deep learning models. Chicho and Sallow (2021) stated in their studies that Keras simplifies neural network development and training through its high-level application programming interface (API). It reduces the intricacies of working with TensorFlow at a lower level, letting developers direct their attention to pay more attention on designing and improving their models without needing to dive into the intricate technical details. This allows users to quickly prototype and deploy models with fewer lines of code, making it particularly appealing for beginners and those looking to experiment rapidly.

According to Yuan (2024), CNN models implemented using PyTorch demonstrate exceptional accuracy in image recognition tasks, achieving rates as high as 99%, although this performance is attributed to the CNN's ability to extract features and learn nonlinear patterns, allowing it to identify complex visual and differentiate categories effectively. The study found that PyTorch have a higher frame rate ensuring better real-time processing capabilities. Furthermore, PyTorch exhibits lower latency due to its dynamic computation graph, which optimizes computation paths during runtime, reduces unnecessary steps, and minimizes processing delays, making it an efficient choice for image recognition tasks.

Kim et al. (2022) made a comparison for Keras and PyTorch for binary image classification using CNN. According to the study, in terms of performance, Keras has shown a higher accuracy, achieving 78% accuracy compared to PyTorch's 70.8%. Additionally, Keras demonstrated a higher F1 score compared to PyTorch, indicating better overall performance in this specific task. It also stated that both frameworks support transfer learning, but they cater to different user needs. Keras simplifies the process with built-in functions for loading and fine-tuning pre-trained models, facilitating easier adoption by beginners in deep learning. In contrast, PyTorch offers more flexibility, allowing researchers to customize the training process and model architecture extensively, which can be advantageous for advanced applications.

Novac et al. (2022) compares the performance of PyTorch and TensorFlow in a depth generation system using stereo images, highlighting key differences in usability, execution times, and accuracy. PyTorch is noted for its user-friendly interface and faster training and execution times, outperforming TensorFlow by an average of 25.5% during training and 77.7% during evaluation, although it achieved slightly lower accuracy (1.16% less) and exhibited less stability in error rates. TensorFlow, while more complex, is favored by experienced users for its flexibility and control, demonstrating better accuracy and stability throughout training, with a lower error rate in depth map generation. The study suggests that the choice between the two libraries significantly impacts design and performance, indicating that PyTorch is better suited for speed-focused tasks, while TensorFlow is preferable for applications prioritizing accuracy.

Therefore, selection of framework has a major effect on the overall design and performance of the model. It is important to select a framework based on the specific needs of task such as in the case of ethnicity classification to ensures an optimal balanced between speed and accuracy, aligning with the project's objectives.

## 2.3 Related Works on Ethnicity Classification

With their capacity to learn complex spatial feature hierarchies, CNNs have greatly advanced image classification tasks, including those involving ethnicity detection. This section reviews CNN architectures used in image recognition, specifically for ethnicity classification.

Wang et al. (2016) introduced a CNN architecture for ethnicity classification, consisting of five layers that includes three convolutional layers and two fully connected layers, culminating in an n-way softmax layer to predict ethnicity probabilities. Model was tested on various datasets, including IDPhotos, CASIA-WebFace, and MORPH-II. It focused on ethnicity groups such as Han, Uyghur, Chinese, and Non-Chinese, achieving accuracy with 99% for most classifications, due to its large and diverse datasets. The model adapts to various face databases by automatically learning relevant features. Despite its accuracy, the model struggles with variations in lighting and head pose and relies on large, potentially biased datasets.

Vo et al. (2018) proposed two models, RR-CNN and RR-VGG, for race classification. RR-CNN is a simple and lightweight model that processes black-and-white images to classify them, while RR-VGG is an improved version VGG-16 that is fine-tuned for better accuracy. Both models are part of a system called the Race Recognition Framework (RRF), which collects data, detects faces, prepares images, and classifies them as "Vietnamese" or "Other" using the VNFaces dataset. VGG model achieved an accuracy of 88.87%, while the CNN model reached 88.64%. These findings demonstrate the effectiveness of VGG models in datasets with clear ethnic distinctions, although challenges remain in generalizing across datasets.

Jilani et al. (2019) explored ResNet architectures that includes ResNet-50, ResNet-101, and ResNet-152 for ethnicity classification of Pakistani faces. These models are designed with deep residual learning, incorporating identity shortcut connections that allow layers to be skipped during training, which enhances performance and reduces computation time without sacrificing accuracy. However, deeper networks like ResNet-152 could demand greater processing power and longer training durations compared to shallower networks. The accuracy achieved by the ResNet architectures is as follows where ResNet-50 achieved an accuracy of 98.8%, ResNet-101 reached the highest accuracy of 99.2%, and ResNet-152 recorded an accuracy of 99.0%.

Ahmed et al. (2020) introduces R-Net, a nine layers CNN with six convolutional layers followed by three fully connected layers, optimized for 40x40 grayscale images. It progressively extracts features, starting with 20 features in the first layer and increasing to 64 features in the last three layers. The architecture incorporates batch normalization and dropout techniques to reduce overfitting, resulting in a total of 8,407,254 trainable parameters. Despite that, model will be limited by the available computing power and the complexity of the model may still lead to overfitting if not managed properly. The study trained using BUPT Equalized Face Dataset and evaluates four ethnicity groups of Caucasian, African, Asian, and Indian, that have achieved accuracies of 77%, 25%, 80%, and 79%, respectively.

Hybrid architectures also show promise by combining multiple models. Ache et al. (2023) combined a VGG16-based CNN with a Support Vector Machine (SVM) classifier for ethnicity classification. SVM is a binary classifier that identifies a hyperplane to separate different classes in the feature space, trained on labeled data. It offers high accuracy and effective feature learning for ethnicity classification but demands significant computational resources and requires careful parameter tuning for SVM component which can be time-consuming. This model was evaluated using two datasets, Iranian Facial Dataset (IFD) and the UTKFace dataset, which include images of Iranian and Asian individuals, respectively. The system achieved 96% accuracy, demonstrating its effectiveness in distinguishing between these ethnic groups.

Alghaili et al. (2023) employed a deep neural network structure called NN4, which generates 128-dimensional embeddings for facial features, classified using an SVM. The use of a Kullback-Leibler (KL) loss function in the study improves representation learning by enabling the model to predict the mean and standard deviation of a Gaussian distribution, enhancing classification performance. However, the integration of the KL loss function can complicate the training process, making it more challenging to optimize the model effectively. The research focused on three specific ethnic groups using CelebFaces Attributes Dataset (CelebA) which are Chinese, Pakistani, and Russian with remarkable accuracy of 97.3%.

Jahan and Mustafa (2024), employs a deep learning approach using a Customized DenseNet121 model for ethnic identification. By connecting each layer to all previous ones, DenseNet121 ensures efficient flow of information and gradients, enhancing feature reuse across the network. It employs bottleneck layers to reduce computational cost, global average pooling to minimize parameters and prevent overfitting, and consists of 121 layers for learning complex patterns. However, the architecture can be complex to implement and requires careful tuning of hyperparameters to achieve optimal performance. The study presents 94% of the model accuracy on a dataset with approximately 4,500 facial images from the Chakma, Marma, and Tripura ethnic groups in the Chittagong Hill district of Bangladesh.

Kalkatawi and Saeed (2024) proposed the model Multi-axis Vision Transformer (MaxViT), designed for ethnicity recognition based on facial images. It classifies six ethnic groups: Asian, Black, Indian, Latino Hispanic, Middle Eastern, and White. The model utilizes a transformer-based architecture, to enhance feature extraction and classification performance. It achieves 77.2% accuracy, demonstrating greater generalization capabilities compared to other models. Besides that, it is recognized as the model with the fewest parameters, which allows for lower computational capacity and improved speed and efficiency. However, implementing transformer-based models like MaxViT may require more sophisticated understanding and resources compared to simpler CNN models due to its complexity.

Choosing the best architecture for ethnicity classification using deep learning depends on factors like model complexity, dataset size, and computational resources. The ideal architecture should balance accuracy, efficiency, and dataset diversity to ensure strong generalization and computational feasibility. In the Table 2.1 below, summary of the deep learning architectures used in previous studies on ethnic classification is listed.

Table 2.1: Summary of Related Work based on Deep Learning Architectures

| **Author(s)** | **Model Architectures** | **Accuracy** | **Ethnicity Groups** | **Advantages** | **Disadvantages** |
|---|---|---|---|---|---|
| Wang et al., 2016 | CNN | 99.50% | Han, Uyghur | Automatically learns effective feature representations | Struggles to generalize under varying conditions like illumination and head pose |
| | | 99.87% | Non-Chinese | | |
| | | 91.60% | Chinese, Non-Chinese | | |
| Vo et al., 2018 | RR-CNN | 88.64% | Vietnamese sub-ethnic group | Learns complex image features for improved accuracy | Limited scope as it only distinguish "Vietnamese" from "Other" |
| | RR-VGG | 88.87% | | | |

| | | | | | |
|---|---|---|---|---|---|
| Jilani et al., 2019 | ResNet-50 | 98.8% | Pakistani | Include identity shortcut connections that allow for skipping layers to improve accuracy | Require more computational resources and time for training |
| | ResNet-101 | 99.2% | | | |
| | ResNet-152 | 99% | | | |
| Ahmed et al., 2020 | R-Net CNN | 77% | Caucasian | Effectively extracts features from images | Limited by computational resources, require downsampling of dataset |
| | | 25% | African | | |
| | | 80% | Asian | | |
| | | 79% | Indian | | |
| Ache et al., 2023 | CNN (VGG16) + SVM | 96% | Iranian and Asian | Improved feature extraction and make reliable predictions across different datasets | Computationally intensive, requiring significant memory and processing power |
| Alghaili et al., 2023 | CNN-NN4 | 97.3% | Chinese, Pakistani, Russian | Utilize Kullback-Leibler loss for effective classification with limited images. | Complexity in loss function integration may complicate training. |
| Jahan and Mustafa, 2024 | Customized DenseNet121, | 94% | Chakma, Marma, Tripura | Better generalization and task-specific accuracy | Complex implementation and resource intensive |
| Kalkatawi and Saeed, 2024 | MaxViT | 77.2% | Asian, Black, Indian, Latino Hispanic, Middle Eastern, White | Require less computation for faster performance | Requires more resources and understanding than simple CNNs |

## 2.4 Transfer Learning

The adoption of transfer learning has shown great potential in enhancing deep learning models, particularly in tasks involving image classification, such as ethnic classification. According to Shin et al. (2016), incorporating transfer learning into the training process can enhance performance on smaller, diverse datasets. They highlighted that pre-trained CNN models significantly improve classification accuracy by transferring learned features from large datasets to smaller, task-specific datasets. This shows the potential of using similar methods for ethnic classification by transferring distinct features learned from large, diverse datasets to specific applications.

Makolo et al. (2023) stated that transfer learning leverages knowledge gained from solving a related problem to address a new domain, reducing both training time and the size of the required dataset. They pointed out that training deep CNNs from scratch is rarely practiced in modern applications due to limited data availability. Instead, they often use pre-trained networks that was previously trained on large datasets to store weights and biases that represent learned features, as initializations or feature extractors for new tasks. The study primarily utilized the EfficientNetB3 pre-trained CNN, which leverages compound scaling to proportionally adjust the network's depth, width, and resolution. Weights from ImageNet were used to initialize this model and employed to extract ethnic salient features from facial images of different ethnic groups. The study also mentions other pre-trained models mentioned include VGG-16 and VGGFace, which were considered for their speed, size, and scalability in various tasks related to ethnicity recognition. Therefore, the utilization of pre-trained models can lead to a more improved accuracy and reduced training times.

### 2.3 Datasets

The advancement of ethnicity classification in deep learning highly dependent on the availability of a large-scale datasets that provide reliable facial images labeled by race or ethnicity. Barbedo (2018) highlighted how the performance of deep learning and transfer learning approaches is affected by the diversity and amount of the data used for training. In the context of ethnicity classification, a rich and varied dataset that captures various ethnic features is essential for developing models that generalize well across different population segments. However, this field faces challenges due to the lack of datasets that comprehensively represent all racial or ethnic groups.

Several datasets have been instrumental in advancing ethnicity classification research. One of the most widely cited datasets is UTKFace, introduced by Zhifei Zhang et al. (2017), which includes 20,000 images with individual label for age, gender, and ethnicity. It categorizes individuals into five ethnic groups: Asian, Black, Indian, White, and Others such as Hispanic, Latino, Middle Eastern. UTKFace is notable for its diverse range of facial variations, including pose, expression, illumination, occlusion, and resolution.

Another significant dataset is the IARPA Janus Benchmark-C (IJB-C), presented by Maze et al. (2018). This publicly available dataset contains 3,531 subjects, with a total of 31,334 images combined with face and non-face images, and 117,542 frames pulled from 11,779 full-motion videos. The dataset represents a wide range of ethnicities from regions such as Americas, Europe (both Western and Eastern), various parts of Africa, the Middle East, South and Southeast Asia, as well as China and East Asia

Region-specific datasets such as IISCIFD, developed by Katti and Arun (2019), contains 1,647 images of North and South Indian faces collected through volunteers and the internet.

BUPT-Balanced dataset, derived from MS-Celeb and introduced by Y. Zhang and Deng (2020), provides 1.3 million images categorized into four racial groups: African, Asian, Caucasian, and Indian.

Another notable dataset is FairFace, proposed by Karkkainen and Joo (2021), consisting 108,192 images distributed across seven ethnic groups: Black, East Asian, Indian, Latino Hispanic, Middle Eastern, Southeast Asian, and White. Data is balanced and was labelled, making it a valuable resource for ethnicity recognition.

Other datasets, like DiveFace, offer 120,000 images grouped into regional clusters, including East Asia, Europe, North America, Latin America, Sub-Saharan Africa, India, and South Asia (Morales et al., 2020).

Earlier studies clearly demonstrated a gap in addressing the classification of a diverse ethnic groups. To overcome the lack of representation in specific contexts, it is crucial to develop custom datasets that focus on the chosen country sub-ethnic groups. These datasets should aim to capture the unique features and characteristics of the country's diverse population, enabling models to achieve higher accuracy and better generalization across all represented groups.

Table 2.2 is the overview of the datasets used in previous studies on ethnic classifications based on their ethnicity, total number of images and the data sources.

Table 2.2 Overview of the datasets

| Dataset Name | Year | Race/Ethnicity | Images | Source |
|---|---|---|---|---|
| UTKFace | 2017 | Asian, Black, Indian, White and Others (Hispanic, Latino, Middle Eastern) | 20K | MORPH, CACD, online resources |
| IJB-C | 2018 | North American, South America, Western Europe, Southwest Africa, East Europe, East Africa-Middle East, South East Asia, India, China, East Asia | 31K | Public, law enforcement databases, social media |
| IISCIFD | 2019 | North Indian and South Indian | 1647 | Volunteers, Internet |
| BUPT-Balanced | 2020 | African, Asian, Caucasian, Indian | 1.3M | MS-Celeb |
| FairFace | 2021 | Black, East Asian, Indian, Latino, Middle Eastern, Southeast Asian, and White | 108K | Flickr, Twitter, newspapers, online resources |
| DiveFace | 2021 | (Japan, China, Korea), (Europe, North America, and Latin America) (Sub-Saharan Africa, India, Bangladesh, Bhutan) | 120K | MegaFace |

## 2.4 Metric Evaluation

The evaluation of ethnicity classification models relies on various performance metrics to assess their accuracy, robustness, and generalizability. These metrics is important to identify the strengths and weaknesses of different models so that they meet the objectives of the classification task.

Accuracy is among the most widely adopted evaluation metrics in classification tasks, defined as the number of correct classifications divided by the total number of samples. Many studies rely on accuracy to compare and determine the best classifying model . However, as highlighted by Hossin and Sulaiman (2015), accuracy can be misleading if there is an imbalance in the datasets or skewed class distributions. In such scenarios, accuracy may favor the majority class, failing to accurately reflect the model effectiveness.

The article also discusses the confusion matrix as a foundational tool for evaluating models. It evaluates classification results per class by sorting predictions into true and false, positives and negatives, to provide information of the model performance and where it struggles, offering insights for further optimization. For instance, Makolo et al., (2023) found that confusion matrix analysis revealed the model difficulty in distinguishing between two classes due to overlapping features, highlighting the need for dataset enhancements and fine-tuning.

Grandini et al., (2020) underscore the importance of Precision, Recall, and F1 Score as key metrics for evaluating classification models. By calculating the ratio of true positives to total predicted positives, precision shows how well the model avoids false positive errors. Recall, or sensitivity, assesses the model's effectiveness in capturing all relevant instances by calculating the ratio of true positives to actual positives. The F1 Score, as the harmonic mean of Precision and Recall, offers a balanced evaluation that will greatly aid in imbalanced data.

A high F1 Score shows that the model can identifies true positives while minimizing false positives, making it a valuable metric for assessing classification performance across diverse applications.

In this project, accuracy and F1-score will be the primary metrics for the performance evaluation of the model, as they provide insights into both correctness and balance. Additionally, confusion matrices will be analyzed to identify misclassifications and refine the model. Given the potential risks of bias in ethnicity classification, fairness metrics will also be explored to ensure equitable performance across all ethnic categories. By combining these evaluation methods, it will help in the development of a reliable model that balances accuracy, fairness, and generalization.

## 2.5 Summary

The development of ethnicity classification models using deep learning hinges on robust datasets, effective architectures, and efficient training methods. While datasets like UTKFace and FairFace offer diverse facial data, a notable gap exists for a country with multi-ethnic populations, highlighting the need for localized datasets. CNNs, including ResNet and DenseNet, excel at learning spatial features and achieving high accuracy. Simpler models like RR-CNN are suitable for lightweight applications, while hybrid approaches combining CNNs with SVM classifiers improve performance for specific ethnic groups. However, distinguishing closely related ethnicities remains a challenge. To address these gaps, it is recommended to develop a custom specific sub-ethnics dataset, explore model architectures for nuanced classification, incorporate transfer leaning, and advanced augmentation approaches to solve the lack of data and improve accuracy.

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

This research methodology focuses on applying deep learning techniques to ethnicity classification, leveraging the VGG-Face architecture within Convolutional Neural Networks (CNNs). The primary objective is to develop a reliable and accurate model capable of categorizing facial images into distinct ethnic groups. By addressing a growing interest in demographic analytics, this study explores robust and reproducible approaches to ethnicity classification, building upon existing methodologies.

The rationale behind adopting the VGG-Face CNN lies in its effectiveness in feature extraction and image classification tasks. As a pre-trained model, VGG-Face has demonstrated remarkable success in facial recognition applications. Its compatibility with transfer learning techniques makes it an ideal foundation for adapting to the specific requirements of ethnicity classification.

Ethnicity classification from images, however, presents unique challenges. Variations in facial features, lighting conditions, angles, and image quality can obscure critical details, complicating the classification process. Furthermore, dataset limitations, such as imbalanced ethnic representation, insufficient diversity, and potential labeling errors or biases, further intensify these difficulties. Addressing these challenges is paramount to developing an accurate and reliable model. This methodology outlines the strategies employed to overcome these obstacles, ensuring a systematic and effective approach to ethnicity classification.

By adopting this structured framework, the study aims to advance the understanding and implementation of ethnicity classification through deep learning, contributing to the broader field of demographic analytics.

## 3.2 Proposed Method

For this study, deep learning techniques, specifically CNNs with VGG architecture, were employed. These networks automatically learn hierarchical feature representations, which capture essential patterns and structures, such as facial features, which are crucial for differentiating between ethnic groups.

### 3.2.1 Convolutional Neural Network

The general architecture for ethnicity classification is built upon CNNs. It is derived from basic neural networks with three types of layers (input, hidden, and output). Every layer is made up of neurons that process outputs from the previous layer, with each neuron operating as a linear classifier by performing mathematical operations like calculating the dot product of the input and its weights, adding a bias value, and applying a non-linear function. This output is then passed to neurons in the next layer.

In traditional neural networks, neurons are fully connected to those in adjacent layers but there are no interconnections between neurons in the same layer. However, this structure becomes inefficient for larger inputs. For example, a 32x32 image would require each neuron in the first hidden layer to handle 1024 weights. If the input size increases to 200x200, a single neuron would need 40,000 weights, which could lead to overfitting and make optimization more challenging.

CNNs address this issue by connecting each neuron to only a small region of the input layer, rather than the entire input. Figure 3.1 shows the diagram for a CNN. This operation is performed in what is known as the convolutional layer (CONV layer). The small region of input that each neuron processes is called the receptive field. The CONV layer stacks neurons in three dimensions: depth, height, and width. The depth represents the number of filters, while the height and width are influenced by the receptive field size, padding, and stride. The output size for height and width can be calculated using the formula:

$$W = H = \frac{(N - F)}{stride} + 1 \qquad \text{Equation (3.1)}$$

where N is the input size, and F is the filter size.

After the convolution operation, a Rectified Linear Unit (ReLU) layer applies a non-linear activation function to the output, making the model capable of learning complex patterns. The ReLU function is monotonic, continuous, and differentiable, which makes it suitable for this purpose.

Following the ReLU layer, a pooling layer is used to reduce the spatial dimensions of the input. This reduces the number of parameters, which helps prevent overfitting and introduces translational invariance to the model. Pooling also ensures the model can handle spatial variations in the input.

Towards the end of the CNN, fully connected (FC) layers take the features learned from the earlier convolutional layers and classify the input image based on these features. Finally, the output of the last fully connected layer was passed through a softmax activation function, which converts the raw output values into probabilities that sum up to 1.

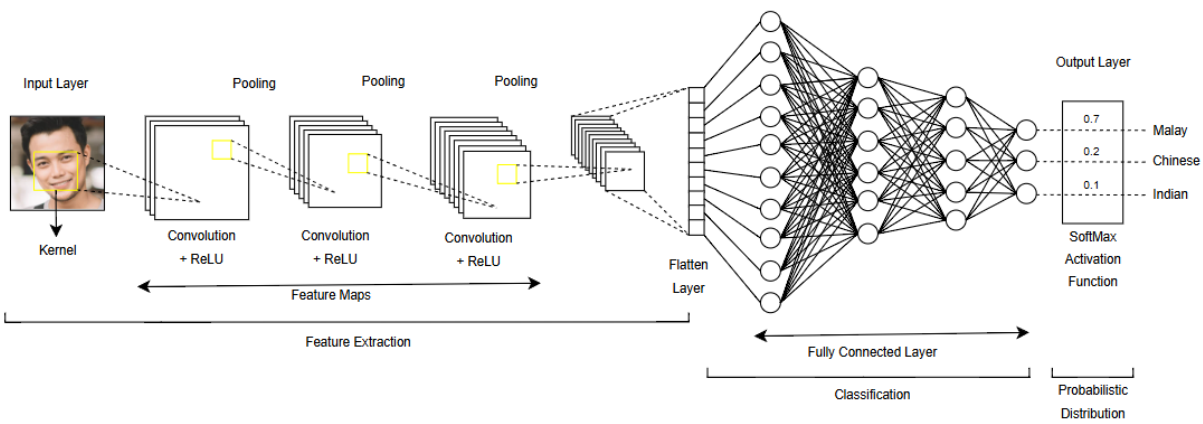



Figure 3.1: CNN diagram architecture

### 3.2.2 Transfer Learning

This section introduces the VGG-Face architecture and explains how transfer learning was applied to develop an efficient ethnicity classification model. The VGG-Face model, a variant of the VGG-16 convolutional neural network (CNN), is pre-trained on a large dataset of facial images, making it highly effective at extracting critical facial features such as the eyes, nose, and mouth which is the key elements for distinguishing ethnicities (Nakada et al., 2017). By leveraging transfer learning, this study avoided the need to train a model from scratch, saving significant computational resources and time. This approach forms the foundation for the implementation, where the pre-trained VGG-Face model is fine-tuned and adapted to classify facial images into three ethnic groups: Malay, Chinese, and Indian.

The VGG-Face model consists of 37 layers organized into six blocks, each designed to extract and process facial features with increasing complexity. The architecture uses 3x3 convolutional filters that slide across the input image (or feature maps) with a stride of 1 and padding of 1, ensuring detailed scanning of every pixel while preserving spatial dimensions.

The first two blocks focus on extracting low-level features such as edges and textures. Each block contains two convolutional layers followed by ReLU activation functions, which introduce non-linearity by setting negative values to zero. A max-pooling layer is applied after the second ReLU layer, reducing spatial dimensions by half while retaining the most important information.

The next three blocks extract higher-level features, such as textures and patterns specific to facial structures. Each block consists of three convolutional layers with ReLU activations, followed by max pooling. This hierarchical process allows the network to progressively learn more complex and abstract representations of the input data.

The final block transitions from feature extraction to decision-making. Here, the feature maps are flattened into a one-dimensional vector and passed through three fully connected layers and a softmax layer. These layers integrate all learned features to produce predictions, with each neuron in a fully connected layer connected to every neuron in the previous layer, ensuring the model considers all extracted features when making decisions.

Figure 3.2 show the VGG-Face architecture. From the diagram, ConvX_Y is where X represents the spatial size of the filter used and Y represents the number of filters used in particular convolutional layer while the feature vector is 4096 dimensional vector (Anwar & Islam, 2017).

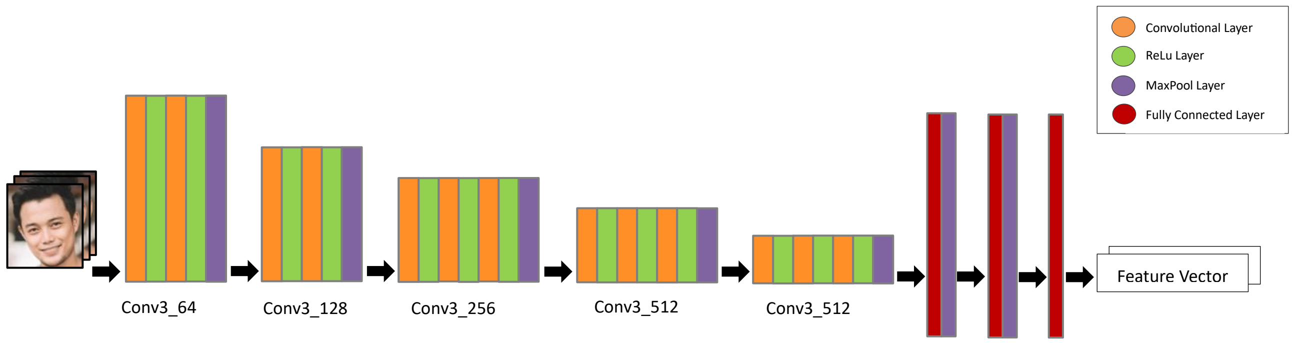



Figure 3.2: VGG Face Network Architecture

Transfer learning was employed to adapt the pre-trained VGG-Face model for the specific task of ethnicity classification. Firstly, the majority of the pre-trained layers were frozen to preserve their ability to extract general facial features, such as edges and textures. Freezing these layers reduced computational costs and minimized the risk of overfitting, especially given the relatively small dataset used in this study.

The original classification head of the VGG-Face model, designed for identifying over 2,600 individual identities, was replaced with a new classification head tailored to this study. The modified classification head consisted of a fully connected layer to process the extracted feature maps, followed by a dropout layer to prevent overfitting by randomly deactivating a fraction of neurons during training, and a softmax layer with three output nodes, each representing one of the target ethnicity classes: Malay, Chinese, and Indian was applied, indicating the likelihood that an input image belonged to each ethnicity.

A few higher-level convolutional layers were unfrozen and fine-tuned using the study's dataset. This step allowed the model to adapt its feature extraction capabilities to ethnicity-specific details, such as variations in facial structures and features that distinguish different ethnic groups. Finally, the modified model was trained on a smaller, labeled dataset of facial images. By combining the general feature extraction capabilities of the pre-trained VGG-Face model with task-specific training, the model effectively learned to classify faces into the three ethnicity categories.

### 3.3 Dataset Preparation

#### 3.3.1 Data Collection

The dataset used in this study was curated from a variety of publicly accessible online sources, including social media platforms, ensuring diverse and representative samples. The primary objective during data collection was to gather enough images from the three target ethnic groups which includes Malay, Chinese, and Indian while minimizing potential biases. Each image was annotated with ethnicity labels and stored for consistent organization and facilitate efficient retrieval during model training and evaluation. To ensure meaningful results, a sufficient number of images per ethnicity is required. However, the dataset remains inherently unbalanced due to the varying representation of ethnic groups.

### 3.3.2 Data Pre-processing

Given the variability in image quality, pose, and content, pre-processing was necessary to ensure uniformity and enhance the performance of the model. The images were resized to a standard dimension of 224 x 224 pixels, aligning with the input requirements of the VGG architecture. This resizing ensures consistency across all input images, reducing variance caused by size discrepancies.

Additionally, a data cleaning process was conducted to remove duplicates, irrelevant images, and instances where unrelated subjects or multiple faces appeared in a single image due to sourcing errors. Duplicate and irrelevant images were identified through a combination of manual inspection and automated tools, ensuring the dataset remained clean and relevant for training. Following this, pixel values were normalized to a range of 0 to 1, enhancing computational efficiency during training.

The dataset was then split into training, validation, and test subsets. This split ensured the model was trained effectively while maintaining separate data for unbiased evaluation and hyperparameter tuning. These pre-processing steps aimed to create a high-quality dataset, optimizing the model's ability to generalize across diverse facial features and ethnic groups.

### 3.3.3 Data Augmentation

For the ethnicity classification task, data augmentation increases the diversity of the training dataset by generating variations of existing images. This improves the model’s generalization, reduces overfitting, and simulates real-world variations in facial images.

One technique used was random rotations, where facial images were rotated by small angles, helping the model recognize faces in various orientations, such as tilted or slightly turned heads. Horizontal flipping, or mirroring images, allowed the model to identify symmetrical features like eyes and jawlines, ensuring ethnic traits remained consistent despite changes in perspective.

Other than that, random zooming and cropping simulated variations in facial sizes, helping the model focus on critical features like eyes, noses, and mouths, even when parts of the face were resized or excluded. Furthermore, an adjustment of the brightness and contrast of the images, it enabled the model to adapt to different lighting conditions, ensuring it could consistently identify ethnic features under varying environments.

### 3.4 Training and Testing

The training and testing process was conducted to assess the model's performance in classifying facial images into three ethnicity categories: Malay, Chinese, and Indian. The process involved two main phases: training a model from scratch, followed by training three additional models using transfer learning.

An initial model was trained from scratch without any pre-trained weights. This served to establish a comparison point for evaluating the benefits of transfer learning. The model was trained using default settings without data augmentation, relying solely on the available dataset.

Subsequently, three models were trained using transfer learning, where a pre-trained architecture was fine-tuned on the ethnicity classification task. These included a baseline model and two fine-tuned variants. Each model differed in terms of regularization strength, dropout rate, and data augmentation strategy. The baseline model was trained without augmentation, while the fine-tuned models incorporated increasingly refined data augmentation pipelines and regularization settings to enhance generalization performance.

All models were trained using the Adam optimizer with a learning rate of 0.001 and a weight decay of 0.0004. The categorical cross-entropy loss function was used, appropriate for multi-class classification tasks. The training was configured with a batch size of 512 and up to 1000 epochs, with early stopping (patience = 30) based on validation loss to prevent overfitting. For Fine-Tuned models, a learning rate scheduler was used to adjust the learning rate when validation loss plateaued. Model checkpoint was also implemented to retain the best-performing model weights.

Once training was complete, the model was evaluated on the testing set to assess its generalization capability using a combination of metrics, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the model's effectiveness in classifying ethnicities. Additionally, a confusion matrix was generated to visualize classification results and identify patterns of misclassification. This analysis aids in fine-tuning the model and addressing areas where performance can be improved.

### 3.5 Summary

In summary, this methodology outlines a systematic approach to ethnicity classification using deep learning techniques. The project leverages the power of CNNs and transfer learning to efficiently train a model capable of handling the challenges of image-based classification. By employing rigorous data preparation, augmentation, and evaluation processes, the methodology ensures the development of a reliable and accurate classification system.

# CHAPTER 4

# MODEL IMPLEMENTATION

## 4.1 Introduction

This chapter outlines the implementation details of the ethnicity classification model, which aims to classify facial images into three ethnic groups: Malay, Chinese, and Indian, using a deep learning approach based on the VGGFace architecture. The primary objective is to leverage transfer learning and fine-tuning to adapt the pre-trained VGGFace model for this specific classification task, achieving robust performance on a custom dataset.

## 4.2 Hardware Specification

The model was developed and trained using Google Colab, a cloud-based platform that offers access to hardware-accelerated resources suitable for deep learning tasks. While initial development used the free tier, Google Colab Pro was also utilized to access enhanced computational resources, including priority access to faster GPUs, longer session durations, and increased RAM availability. These improvements significantly reduced training time, especially during multiple experiments and fine-tuning phases, by enabling larger batch sizes and faster data processing. The specifications are summarized in Table 4.1.

Table 4.1: Hardware and Software Specifications

| Component | Specification |
|---|---|
| **GPU** | T4 GPU (Free), NVIDIA L4 GPU 22.5 GB VRAM (Pro) |
| **RAM** | 12.7 GB (T4 GPU), 53.0 GB (L4 GPU) |
| **Operating System** | Ubuntu (via Google Colab) |
| **Environment** | Google Colab, Jupyter Notebook |
| **Libraries and Framework** | TensorFlow/Keras, OpenCV, NumPy, Matplotlib |

These resources provided sufficient computational power to handle the dataset and training requirements of the VGGFace model, with TensorFlow/Keras serving as the primary framework for model development.

### 4.3 Dataset Collection

#### 4.3.1 Raw Data Acquisition

The dataset for this project comprises facial images representing three ethnic groups: Malay, Chinese, and Indian. Images were primarily sourced from Google Images and various websites using web scraping approach. Custom Python scripts were developed using Selenium, enabling the automatic searching and downloading of images based on relevant keywords associated with the three target ethnicities. In addition to this automated collection, the dataset was further supplemented through manual selection, including publicly available images of celebrities and athletes, to ensure representative samples across all classes. For the Indian ethnic group, an additional dataset was incorporated from a publicly available repository on GitHub by Katti and Arun (2019). Ethical considerations were addressed by using only publicly available images. The initial number of images collected per ethnic group is shown in Table 4.2.

Table 4.2: Initial Dataset Size

| Ethnic Group | Number of Images |
|---|---|
| **Malay** | 553 |
| **Chinese** | 570 |
| **Indian** | 528 |

#### 4.3.2 Data Preprocessing

Before training the model, a comprehensive data preprocessing phase was conducted, which included cleaning the dataset preparing the images for training, encoding the ethnicity labels into a machine-readable format and data splitting.

Firstly, to ensure that the dataset's quality, several cleaning steps were applied to remove unsuitable images. These steps included:

1. Removal of low-quality or blurry images using a Laplacian variance-based blurriness detection algorithm.
2. Detection and removal of duplicate images to avoid redundancy.

3. Face detection and cropping using the Multi-Task Cascaded Convolutional Neural Network (MTCNN) to ensure each image contains exactly one face.

4. Manual review to verify the quality and relevance of the images.

After cleaning, the dataset was balanced to 460 images per ethnic group, resulting in a total of 1,380 images. This balanced dataset ensured equitable representation of each class during training.

Table 4.3: Dataset Size after Cleaning

| Ethnic Group | Number of Images |
|---|---|
| **Malay** | 460 |
| **Chinese** | 460 |
| **Indian** | 460 |

Next, to prepare the dataset for the VGGFace model, several preprocessing steps were applied to ensure consistency and optimize training efficiency. Each image was resized to 224 × 224 pixels which is the input size required by VGGFace. Images were converted into NumPy arrays, with pixel values stored in a DataFrame column named pixels. These arrays were reshaped into a 4D tensor with the format (N, 224, 224, 3), where N is the number of samples, and 3 represents the RGB channels. Pixel normalization was performed by scaling pixel values to the range [0,1] through division by 255. This step stabilizes gradient updates and mitigates the risk of exploding or vanishing gradients during training. The preprocessing steps are summarized as follows:

1. Resize images to 224×224 pixels.

2. Convert images to NumPy arrays and reshape into a 4D tensor.

3. Normalize pixel values to [0,1].

Then, label encoding was applied to convert the categorical ethnicity labels into integer values. Each unique ethnicity was identified and replaced with a corresponding integer, starting from 1. To ensure compatibility with the model training pipeline, the encoded labels were cast to 32-bit integers. Table 4.4 presents the mapping of ethnicity labels to their encoded integer values.

Table 4. 4: Ethnicity Labels Mapping

| Ethnic Group | Encoded Integer Values |
|---|---|
| **Indian** | 1 |
| **Malay** | 2 |
| **Chinese** | 3 |

Finally, the dataset was split into training, validation, and testing sets to facilitate model development and evaluation. Initially, the 1,380 images were divided into 80% training (1,104 images) and 20% testing (276 images). The test set was reserved as unseen data to evaluate the model's performance. From the training set, 12% (133images) was further allocated to a validation set to monitor training progress and prevent overfitting. The final split is shown in Table 4.5.

Table 4.5: Dataset Split

| Set | Number of Images |
|---|---|
| **Training** | 971 |
| **Validation** | 133 |
| **Testing** | 276 |

### 4.3.3 Data Augmentation

To enhance the generalization capability of the model in ethnicity classification, data augmentation was applied to the training images of the fine-tuned models. Data augmentation artificially increases dataset variability through image transformations, helping the model learn invariant facial features and reducing overfitting. This also allows the model to handle real-world variations such as lighting conditions, head orientation, image blur, and scale changes.

Augmentation was implemented using the Albumentations library, a fast and flexible tool for image transformation in deep learning workflows. Two augmentation pipelines were developed to improve model generalization, each applied to different fine-tuned models.

Pipeline 1, used in Fine-Tuned 1, consisted of horizontal flipping, rotation, brightness and contrast adjustments, geometric transformations (shifting, scaling, and rotating), Gaussian blur, and random cropping, with all images subsequently resized to 224×224 pixels. Pipeline 2, used in Fine-Tuned 2, followed the same structure but with a reduced rotation limit to apply more conservative transformations while maintaining consistency in the other augmentation parameters.

Table 4.6: Augmentation Parameters

| Transformation | Limit | Probability (p) |
|---|---|---|
| **Horizontal Flip** | - | 0.5 |
| **Rotate** | 15° (P1), 10° (P2) | 0.3 |
| **Random Brightness/Contrast** | brightness =0.2, contrast = 0.2 | 0.3 |
| **Shift, Scale, Rotate** | shift = 0.1, scale = 0.1, rotate= 10 | 0.3 |
| **Gausssian Blur** | blur = (3, 7) | 0.2 |
| **Random Crop** | height = 200, width = 200 | 0.2 |

Visualizations of each augmentation were shown in Figure 4.1 to qualitatively verify their effects and confirm that key facial features remained recognizable post-transformation.

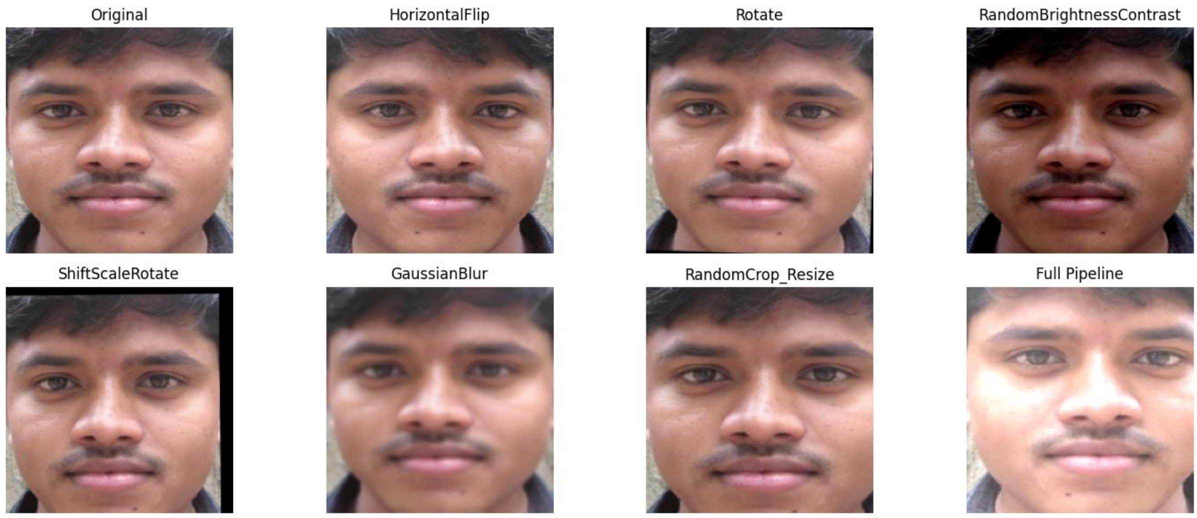



Figure 4.1: Visualization of Data Augmentation

### 4.4 Model Training without Transfer Learning

To assess the effectiveness of transfer learning in this project, an initial model was trained entirely from scratch, without incorporating any pre-trained weights. In this setup, the entire network architecture was initialized with random weights and learned to extract features and perform ethnicity classification solely based on the provided dataset. This approach allowed the model to develop representations independently, without relying on prior knowledge from external datasets. The performance of this model served as a reference point to evaluate the impact of transfer learning, providing valuable insight into how much pre-trained features contribute to improving classification accuracy and generalization in this task.

### 4.5 Model Fine-Tuning using Transfer Learning

The proposed model for ethnicity classification is built on top of the VGG-Face model, a deep convolutional neural network pre-trained for face recognition. The architecture is modified using transfer learning to adapt it for a 3-class ethnicity classification task (Malay, Chinese, Indian). The original VGG-Face consists of multiple convolutional layers organized into blocks, followed by fully connected layers for classification. It takes input images of size 224 × 224 × 3 and outputs predictions over 2,622 facial identities.

To adapt the model for ethnicity classification, the original architecture was preserved up to the final convolutional layers. All layers except the last 6 layers were frozen to retain the rich facial features learned from large-scale face recognition tasks.

```
# Freeze all layers of VGG-Face except last 7 one
for layer in model.layers[:-6]:
    layer.trainable = False
```

Figure 4.2: Code snippet for freezing layers

Then, the final convolutional layer was replaced with a new 1×1 convolution layer having 3 output filters, corresponding to the 3 ethnic classes. Modifications were made to the output to allow classification into three classes: Malay, Chinese, and Indian. A softmax activation was applied to produce class probabilities.

```
num_of_classes = 3

# Modify the existing model to create the new output
base_model_output = Convolution2D(num_of_classes, (1, 1),
name='predictions')(model.layers[-4].output)
base_model_output = Flatten()(base_model_output)
base_model_output = Activation('softmax')(base_model_output)
```

Figure 4.3: Code snippet for modifying the output layer into 3 classes

To optimize the VGG-Face model for ethnicity classification, a multi-stage fine-tuning approach was adopted. Three models were developed and evaluated, each incorporating different levels of regularization and data augmentation strategies as shown in Table 4.7.

Table 4.7: Model Training Configuration

| Model | Data Augmentation | L2 Regularization | Dropout Rate |
|---|---|---|---|
| **Baseline** | None | 0.01 | 0.5 |
| **Fine-Tuned 1** | Pipeline 1 | 0.001 | 0.3 |
| **Fine-Tuned 2** | Pipeline 2 | 0.0001 | 0.35 |

All models were trained using the Adam optimizer with a learning rate of 0.001 and weight decay of 0.0001, ensuring stable convergence and effective regularization during training. L2 Regularization adds a penalty to the loss function proportional to the square of the weights. This discourages the model from learning overly complex patterns by keeping the weights small, thus helping to prevent overfitting, especially when training on relatively small datasets. For dropout, it will randomly deactivate a fraction of neurons during each training step. This forces the network to learn redundant representations and prevents co-adaptation of neurons, further reducing the risk of overfitting. Different dropout rates were tested across models to find the optimal balance between model complexity and generalization.

## 4.6 Training, Validation and Testing

### 4.6.1 Training Setup

The model was trained using the following configuration in Table 4.6.

Table 4.8: Training Configuration

| Configuration | Description |
|---|---|
| **Loss Function** | Categorical cross-entropy |
| **Optimizer** | Adam optimizer |
| **Learning Rate** | 0.001 |
| **Epochs** | 1000 |
| **Batch Size** | 256 |
| **Early Stopping (Patience)** | 30 epochs |
| **Checkpointing** | Save model weights |
| **Learning Rate Reducer** | Factor=0.5, Patience=10, Min_lr=0.000001 |

The model was trained using the categorical cross-entropy loss function, which is commonly used for multi-class classification problems. The formula for this loss function is:

$$Loss = \sum_{i=1}^{N} y_i log(\hat{y}_i) \qquad \text{Equation (4.1)}$$

where $\hat{y}_i$ represents the predicted probability of the correct class, $y_i$ as the true label, and N as the number of classes. To optimize the model, Adam optimizer was used due to its efficiency and ability to adapt the learning rate during training. The learning rate was set to Adam optimizer default learning rate.

```
race_model.compile(loss='categorical_crossentropy'
                   , optimizer=keras.optimizers.Adam()
                   , metrics=['accuracy']
                  )
```

Figure 4.4: Code snippet for loss function and optimizer used

To improve training efficiency and generalization, batch training with a size of 256 was used, taking advantage of available GPU resources to enable faster training and more stable gradient updates. In each epoch, the model updated its weights based on randomly selected subsets (batches) of the training data, simulating the behavior of stochastic gradient descent (SGD). A larger batch size like 256 helps speed up training and can improve convergence, especially when supported by a higher-end GPU. For environments with limited memory, a smaller batch size (e.g., 64) may be used to reduce memory consumption at the cost of slower training.

The training was run for up to 1000 epochs, allowing sufficient iterations for convergence, but an early stopping mechanism was implemented to prevent overfitting. The training would stop early if the validation loss did not improve after 30 consecutive epochs (patience), indicating that the model had likely converged.

A model checkpoint was used to save the best-performing model which is the one with the lowest validation loss during training. This ensured that the final model retained the best weights even if overfitting occurred later.

```
# Batch training loop with early stopping and checkpointing
for i in range(epochs):
    batch_idx = np.random.choice(train_x.shape[0], size=batch_size)
    score = race_model.fit(
        train_x[batch_idx], train_y[batch_idx],
        epochs=1,
        validation_data=(val_x, val_y),
        callbacks=[checkpointer]
    )

    val_loss = score.history['val_loss'][0]
    if val_loss < best_loss:
        best_loss = val_loss
        last_improvement = 0
    else:
        last_improvement += 1

    if last_improvement == patience:
        break
```

Figure 4.5: Code snippet for batch training loop with early stop and model checkpoint

### 4.6.2 Validation Strategy

The validation strategy played a critical role in monitoring and improving the model's performance during training. A dedicated validation set consisting of 133 images was set aside from the main dataset and was not used during training. This ensured that the model was evaluated on data it had never seen, providing a reliable indication of its ability to generalize to new, unseen examples.

During training, validation loss and validation accuracy were continuously monitored after each epoch. These metrics were used to assess whether the model was learning useful features or simply memorizing the training data (overfitting). To further support this, an early stopping mechanism was implemented with a patience value of 30 epochs. If no improvement in validation loss was observed for 30 consecutive epochs, training was automatically halted to prevent overfitting and save computational resources.

This strategy allowed for dynamic tuning of training behavior that helped identify the optimal number of training epochs. It also prevented excessive training that could degrade performance on unseen data and provided a checkpointing mechanism to save the best-performing model based on validation performance.

Furthermore, by tracking both training and validation losses over time, trends were analyzed to adjust other hyperparameters like learning rate or batch size if necessary. This ensured the final model maintained a balance between underfitting and overfitting, thereby improving generalization performance on the test set and on out-of-distribution (external) samples.

**4.6.3 Testing**

The final model was evaluated on the test set (276 images), which was kept separate from the training and validation sets. The test set performance, including metrics such as accuracy and loss, will be detailed in Chapter 5. No cross validation was performed in this study due to computational constraints and the balanced nature of the dataset.

```
# Test and Validation Perfomance
test_perf = race_model.evaluate(test_features, test_target.values,
```

```
verbose=1)
validation_perf = race_model.evaluate(val_x, val_y, verbose=1)
#Check model is robust
abs(validation_perf[0] - test_perf[0]) < 0.01
```

Figure 4.6: Code snippet for calculating accuracy and model checking

To evaluate the generalization capability of the trained model, predictions were also made on images that were not part of the training, validation, or testing datasets. These out-of-sample images were manually collected and represented real-world faces of individuals from the three target ethnicities (Malay, Chinese, Indian). The primary purpose of this testing approach was to simulate real deployment scenarios, where the model must accurately classify unseen data. These images varied in background, lighting conditions, and facial orientation to test the model's robustness.

### 4.7 Prototype Development

In order to validate and demonstrate the practical applicability of the ethnic classification model, a real-time prototype was developed using OpenCV and TensorFlow Keras. This prototype enables real-time face detection and ethnic classification through a webcam interface, providing immediate visual feedback.

The prototype begins by loading the deep learning model using TensorFlow's Keras API. This model was previously trained to classify faces into one of three ethnic groups: Indian, Malay, and Chinese. A dictionary was used to map the model's predicted class indices to the corresponding ethnicity labels. For detecting faces in real-time from the webcam feed, the system employs OpenCV's pre-trained Haar Cascade Classifier. This method efficiently locates facial regions within each video frame. The classifier detects frontal faces using grayscale conversion and bounding box estimation. Each detected face is cropped and pre-processed before being passed to the deep learning model. This includes resizing to 224 × 224 pixels, normalizing pixel values to the [0,1] range and expanding the array's dimensions to match the expected input shape to ensure compatibility with the model's training configuration.

During the live video stream, the system processes each detected face in real-time. The model makes predictions for each face, returning probabilities for each class. The class with the highest probability is selected as the predicted ethnicity. Additionally, the system calculates the prediction confidence (as a percentage). The predicted ethnicity and its confidence score are overlaid on the video feed. A bounding box is drawn around each detected face, and the corresponding label is displayed above the box. The OpenCV GUI window displays the annotated video stream, allowing users to view classification results live. The prototype runs continuously and can be stopped by pressing the 'q' key or through keyboard interruption. Proper exception handling and resource cleanup ensure the webcam is released and the window is closed gracefully.

# CHAPTER 5

# RESULT AND DISCUSSION

## 5.1 Introduction

This chapter presents the results of the ethnicity classification model based on the VGGFace architecture and discusses their performance and implications. The primary goal is to analyze the model's performance in classifying three ethnic groups using quantitative metrics such as accuracy and confusion matrices. The chapter evaluates the impact of different training configurations during fine tuning and compares models with and without transfer learning. Additionally, it discusses observed patterns, limitations, and potential improvements to enhance the model's robustness and applicability.

## 5.2 Training Results

The VGGFace model was trained in four configurations, without transfer learning and with transfer learning with different fine-tuning training: Baseline, Fine-Tuned 1, and Fine-Tuned 2. Training was conducted over 1,000 epochs with a batch size of 256, using the Adam optimizer and categorical cross-entropy loss. Early stopping with a patience of 30 epochs was applied to prevent overfitting. The training and validation accuracy and loss were monitored to assess the model's learning behavior. Table 5.1 provides the training results of all the models. A more detailed explanation will be provided in the following subsection.

Table 5.1: Training Results of All Models

| Model | Val Loss | Test Loss | Val Accuracy | Test Accuracy | Loss Difference – Robust Check (Val - Test Loss < 0.05) |
|---|---|---|---|---|---|
| No Transfer Learning | 1.1035 | 1.1007 | 33.08% | 33.33% | 0.0032 |
| Baseline | 0.6561 | 0.6794 | 89.47% | 88.41% | 0.0168 |
| Fine-Tuned 1 | 0.4226 | 0.4345 | 91.73% | 89.86% | 0.0276 |
| Fine-Tuned 2 | 0.3313 | 0.3501 | 91.73% | 91.30% | 0.0420 |

### 5.2.1 Learning Curves

The training and validation loss curves were plotted to visualize the model's performance and provide insights into how well the model is minimizing error on both the training and validation datasets across epochs. A consistently decreasing trend in both losses indicates good learning and generalization. A decreasing training loss with an increasing validation loss suggests overfitting, where the model memorizes training data but fails to generalize. A high training loss with a low validation loss point to underfitting, which can be due to poor model capacity or under-trained to capture data patterns.

Table 5.2: Test and Validation Loss

| Model | Val Loss | Test Loss | Loss Difference – Robust Check (Val - Test Loss < 0.05) |
|---|---|---|---|
| No Transfer Learning | 1.1035 | 1.1007 | 0.0032 |
| Baseline | 0.6561 | 0.6794 | 0.0168 |
| Fine-Tuned 1 | 0.4226 | 0.4345 | 0.0276 |
| Fine-Tuned 2 | 0.3313 | 0.3501 | 0.0420 |

During the first epoch for model training without transfer learning, the training loss starts extremely high and sharply drops to approximately 1.0, indicating rapid learning from initially poor or random weight initialization, while the validation loss begins at about 1.1 and stays nearly flat. From epoch 1 to 4, both losses remain low and stable with minimal changes, suggesting early model convergence, potential underfitting due to halted learning, a very low learning rate, or insufficient data to drive further improvements. Figure 5.1 shows the training and validation loss curve over epochs for this model without transfer learning.

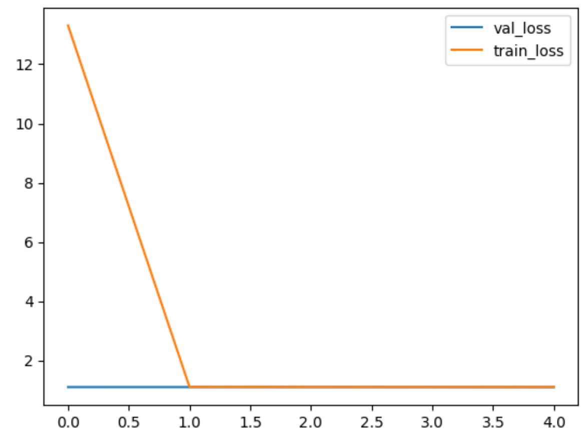



Figure 5.1: Train and Val Loss Curve over Epochs (Without Transfer Learning)

For the training and validation loss curves of the baseline model shown in Figure 5.2, initially, the training loss starts above 3.0, indicating poor initial model performance, while the validation loss begins around 1.5 which is relatively lower possibly due to strong regularization or low initial model capacity. In the first 10 epochs, both losses drop rapidly, reflecting effective early learning as the model captures basic data patterns. After this phase, the training loss stabilizes around 0.9–1.0 and the validation loss around 0.75–0.8, indicating a plateau. The validation loss remains consistently lower than the training loss, which may be due to dropout, strong regularization, or techniques like early stopping or model averaging. Overall, the model converges well within 10–15 epochs, and the lower validation loss suggests a lack of overfitting, though it may also hint at underfitting or regularization effects.

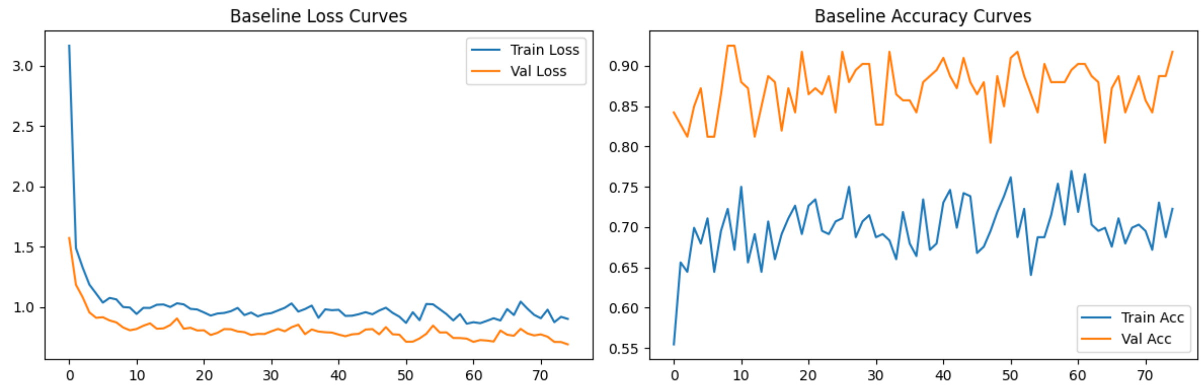



Figure 5.2: Train and Val Loss Curve over Epochs (Baseline)

The loss curves for Fine-Tuned 1 show a strong improvement over the baseline model. Training begins with a lower loss (~1.15) and validation loss (~0.75), indicating effective initialization, possibly from transfer learning or fine-tuning. Both losses drop sharply within the first 10 epochs, reflecting rapid early learning, and then stabilize. Training loss between 0.45–0.6 and validation loss between 0.42–0.6. Notably, the validation loss remains consistently lower or comparable to the training loss, suggesting minimal overfitting and robust generalization. While minor validation loss fluctuations appear around epochs 25 and 50, the overall trend is downward and stable. Compared to the baseline, Fine-Tuned 1 achieves lower final losses, higher stability, and improved generalization. These results demonstrate that the model learns faster and more efficiently, likely due to fine-tuning strategies.

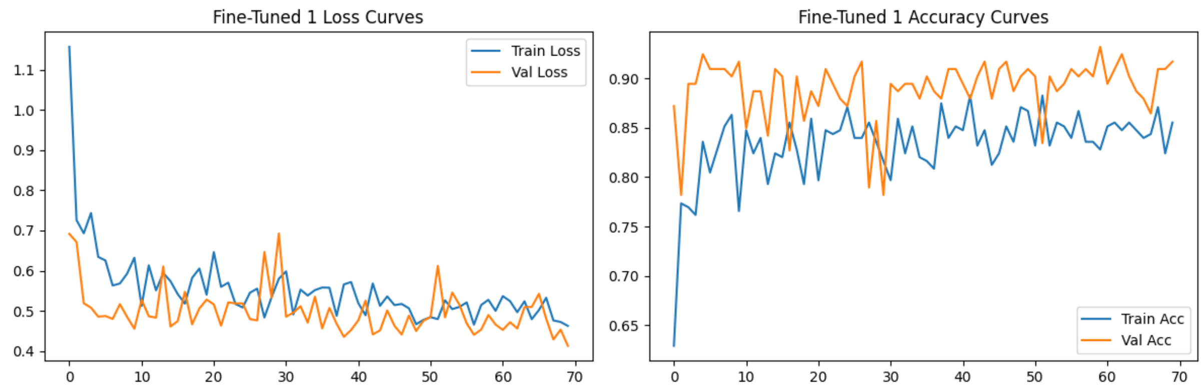



Figure 5.3: Train and Val Loss Curve over Epochs (Fine-Tuned 1)

The loss curves for Fine-Tuned 2 indicate the most optimized performance among all models evaluated. Training begins at a relatively low loss (~0.93), while validation starts even lower (~0.55), outperforming both the baseline and Fine-Tuned 1 from the outset, suggesting effective initialization or reuse of prior knowledge. Rapid improvements occur within the first five epochs, with validation loss dipping below 0.35, highlighting strong early generalization. Afterward, training loss gradually settles between 0.38–0.45, while validation loss remains consistently lower, typically ranging from 0.29–0.38. A minor spike at epoch 23 is quickly corrected, reinforcing the model's stability. From around epoch 10 onward, both curves plateau, indicating convergence with minimal fluctuation. Compared to the baseline and Fine-Tuned 1, Fine-Tuned 2 achieves the lowest final training (~0.4) and validation (~0.29) losses, shows no signs of overfitting, and exhibits very high consistency, making it the most effective model to date in terms of loss minimization and generalization.

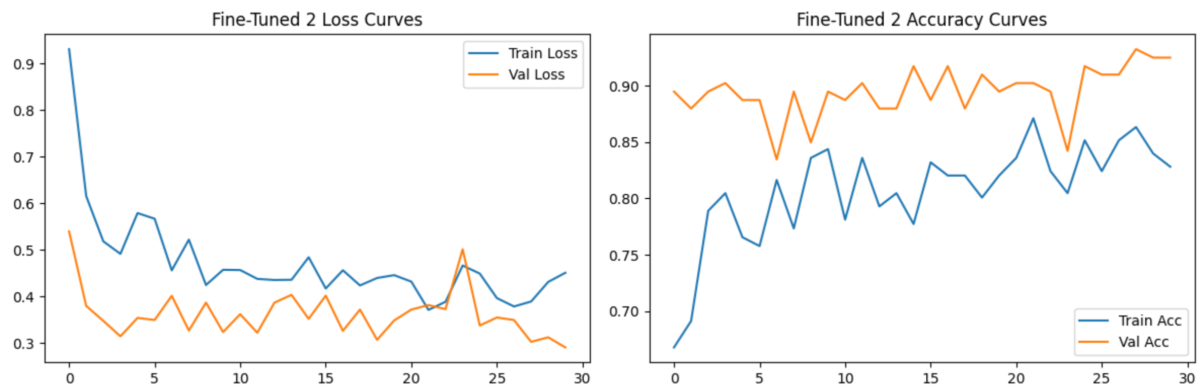



Figure 5.4: Train and Val Loss Curve over Epochs (Fine-Tuned 2)

### 5.2.2 Performance Evaluation

The models were evaluated on the test set (276 images) to assess performance on unseen data. Table 5.3 summarizes the test accuracy for each configuration, including a baseline model without transfer learning.

Table 5.3: Test Set Accuracy Performance

| Model | Val Accuracy | Test Accuracy |
|---|---|---|
| No Transfer Learning | 33.08% | 33.33% |
| Baseline | 89.47% | 88.41% |
| Fine-Tuned 1 | 91.73% | 89.86% |
| Fine-Tuned 2 | 91.73% | 91.30% |

Among the metrics, accuracy provides the most straightforward measurement, representing the proportion of correctly classified images out of the total predictions. It is calculated as follows:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ od\ predictions} \times 100 \quad \text{Equation (5.1)}$$

The accuracy results highlight the significant impact of transfer learning and fine-tuning on model performance. The model without transfer learning performs poorly, achieving only 33.08% validation accuracy and 33.33% test accuracy—barely above random guessing—indicating that training from scratch is ineffective for this task. The baseline model, which incorporates transfer learning, shows a dramatic improvement, achieving 89.47% on the validation set and 88.41% on the test set, confirming the value of leveraging pretrained knowledge. Fine-Tuned 1 builds on this by slightly improving both validation (91.73%) and test accuracy (89.86%), suggesting that additional tuning enhances generalization. Fine-Tuned 2 matches Fine-Tuned 1 in validation accuracy (91.73%) but achieves the highest test accuracy at 91.30%, reflecting superior generalization to unseen data. Overall, Fine-Tuned 2 demonstrates the best balance between validation and test performance, marking it as the most effective model configuration among those tested.

The precision, recall, and F1-score by ethnicity were also recorded for each of the model trained with transfer learning as shown in Table 5.4. The results indicate that all models perform relatively well across ethnic groups, but there are notable differences in how each model handles class-specific precision and recall. The Baseline model shows strong and balanced performance, particularly for Malay (F1-score: 0.94), with high precision and recall across all three groups. However, it slightly underperforms for Chinese (F1-score: 0.84) due to lower recall (0.82), suggesting some missed predictions for this group.

Fine-Tuned 1 demonstrates improved recall for Indian (0.95) and consistent performance for Malay (F1-score: 0.95), indicating robust recognition for these groups. However, its performance on Chinese drops noticeably, with a lower recall (0.74) and F1-score (0.81), suggesting the model struggles more to correctly identify Chinese instances.

Fine-Tuned 2 offers the most balanced performance overall. It improves Indian precision to 0.89 and maintains high recall (0.92), leading to the best F1-score (0.91) for this group. For Malay, it achieves the highest recall (0.98) while maintaining strong precision (0.91), resulting in a stable F1-score of 0.94. It also improves Chinese performance over Fine-Tuned 1, with higher precision (0.90) and better recall (0.80), lifting the F1-score to 0.85.

Overall, Fine-Tuned 2 demonstrates the best balance of precision and recall across all ethnicities, indicating that it not only generalizes well but also handles class imbalances or subtle differences in group features more effectively than previous models.

Table 5.4: Precision, Recall and F1-Score

| Model | Ethnicity | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Baseline | Indian | 0.85 | 0.93 | 0.89 |
| | Malay | 0.97 | 0.92 | 0.94 |
| | Chinese | 0.86 | 0.82 | 0.84 |
| | | | | |
| Fine- Tuned 1 | Indian | 0.81 | 0.95 | 0.87 |
| | Malay | 0.95 | 0.95 | 0.95 |
| | Chinese | 0.89 | 0.74 | 0.81 |
| | | | | |
| Fine-Tuned 2 | Indian | 0.89 | 0.92 | 0.91 |
| | Malay | 0.91 | 0.98 | 0.94 |
| | Chinese | 0.90 | 0.80 | 0.85 |

Furthermore, the confusion matrices for the all the models trained were presented to show the classification performance per ethnic group. The confusion matrix in Figure 5.5 clearly illustrates that the model without transfer learning is ineffective, failing to learn any discriminative features. All samples are predicted as Chinese, leading to poor classification performance across the board. It highlights the necessity of transfer learning or other advanced techniques to achieve meaningful results on this task.

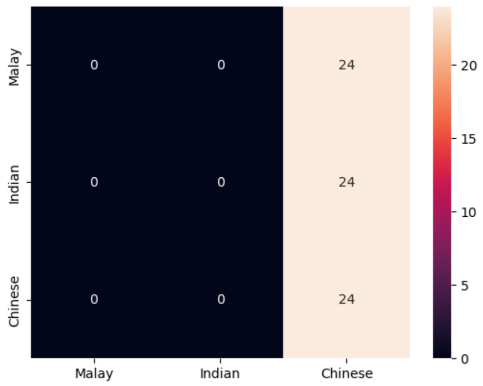



Figure 5.5: Confusion Matrix for Model Without Transfer Learning

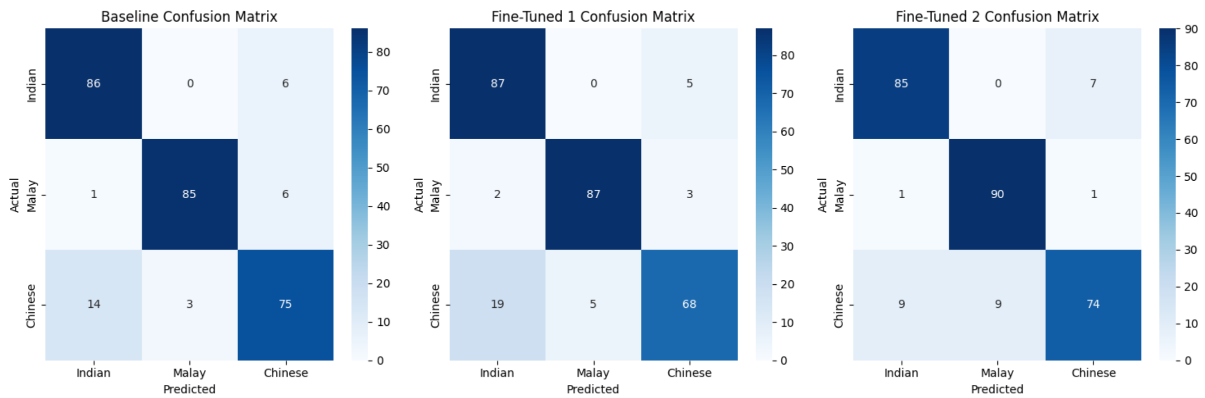



Figure 5.6: Confusion Matrices for Model with Transfer Learning

The confusion matrix for the Baseline model demonstrates strong overall classification performance, particularly for the Indian and Malay classes. The model correctly classifies 86 out of 92 Indian samples and 85 out of 92 Malay samples. However, performance on the Chinese class is slightly weaker, with only 75 correct predictions and 17 misclassifications—14 labeled as Indian and 3 as Malay. This suggests that while the model has learned to distinguish Indian and Malay features quite well, it is somewhat less reliable in identifying Chinese instances, possibly due to overlapping feature representations.

In the Fine-Tuned 1 model, performance on Indian and Malay samples remains strong, with 87 and 87 correct classifications, respectively. However, the model shows a noticeable drop in Chinese class accuracy, correctly predicting only 68 out of 92 samples. A total of 24 Chinese instances are misclassified—19 as Indian and 5 as Malay—indicating a decline in the model's ability to differentiate Chinese features. This could be due to the model overfitting to the Indian and Malay classes during fine-tuning or under-representing certain characteristics of the Chinese class.

The Fine-Tuned 2 model exhibits the most balanced and consistent performance across all three classes. Malay classification reaches its highest level of accuracy with 90 correct predictions and only two misclassifications (one each as Indian and Chinese), indicating a refined understanding of this class. Indian classification remains strong with 85 correct predictions, slightly lower than in previous models but still robust. Importantly, Chinese classification improves compared to Fine-Tuned 1, with 74 correct predictions and fewer errors—only 9 misclassified as Indian and 9 as Malay. This suggests that Fine-Tuned 2 effectively recovers performance on the Chinese class while maintaining high accuracy across the board.

Overall, while all three models perform well, Fine-Tuned 2 demonstrates the best generalization and balance, minimizing misclassifications and offering improved consistency across all ethnic groups. This confirms its superiority over the baseline and earlier fine-tuned version in terms of both robustness and fairness across class predictions.

To provide qualitative insight into the model's predictions, Figure 5.7 displays examples of correct predictions for the three different classes, while Figure 5.8 shows examples of incorrect classifications. In Figure 4.6, the misclassification highlights the challenges in distinguishing facial features among certain ethnic groups.

| | | |
|---|---|---|
|  |  |  |
| Actual: Indian<br>Predicted: Indian | Actual: Malay<br>Predicted: Malay | Actual: Chinese<br>Predicted: Chinese |

Figure 5.7: Examples for Correct Predictions

| | | |
|---|---|---|
|  |  |  |
| Actual: Chinese<br>Predicted: Malay | Actual: Malay<br>Predicted: Indian | Actual: Chinese<br>Predicted: Malay |

Figure 5.8: Examples for Wrong Predictions

## 5.3 Discussion

### 5.3.1 Class-wise Performance Analysis

The confusion matrices reveal that the Indian class was classified with the highest accuracy across all models, followed closely by the Chinese class, while the Malay class was the most challenging to predict accurately. This pattern suggests possible variability within the Malay class, or overlapping facial characteristics between Malay and Chinese individuals, which may arise from historical and regional intermixing. Such overlap can make it more difficult for the model to distinguish between these groups using only facial features. Furthermore, the diversity within each ethnic group adds complexity to the task, as intra-class variation can sometimes exceed inter-class differences. Misclassifications may also reflect the model's reliance on superficial or low-level visual cues, such as skin tone or general face shape, rather than more nuanced, identity-related features that are crucial for accurate classification.

### 5.3.2 Impact of Transfer Learning

The application of transfer learning significantly enhanced the model's performance. By leveraging pre-trained weights from the VGGFace model, the classifier benefited from robust feature extraction in the early layers, which are already tuned to detect fundamental facial patterns like edges, contours, and textures. This enabled the model to converge faster and generalize better, particularly when compared to the baseline model trained from scratch. Moreover, fine-tuning selected layers of the pre-trained model allowed it to adapt more effectively to the specific task of ethnicity classification.

### 5.3.4 Additional Observations

Model performance was found to be sensitive to both image quality and dataset diversity. High-resolution images with clearly visible facial features generally resulted in more accurate predictions, while blurred, occluded, or low-quality images often led to misclassifications. The reliance on datasets composed primarily of celebrity and athlete images may have introduced a bias toward idealized or stylized facial appearances, which does not necessarily reflect the broader population. This could limit the model's real-world applicability and generalization. Additionally, the lack of data augmentation such as variations in lighting, pose, and background reduced the model's exposure to natural variability, likely contributing to some overfitting and reduced robustness in more challenging or unconstrained environments.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## 6.1 Introduction

This chapter summarizes the key findings and outcomes of the project, reviews the extent to which the project objectives were achieved, discusses existing limitations, and proposes directions for future improvements and research. The project aimed to develop an ethnicity classification model using deep learning techniques. Through transfer learning and performance evaluation, the project provides valuable insights into the practical application of deep learning in facial analysis and demographic classification.

## 6.2 Achievement

Table 6.1 below summarizes the primary objectives of this project as stated in Chapter 1, alongside the corresponding achievements. Each achievement reflects how the project successfully addressed its respective objective through the application of deep learning techniques, dataset development, and model evaluation strategies.

Table 6.1: Summary of Project Objectives and Corresponding Achievements

| Objective | Achievement |
|---|---|
| 1. Develop a deep learning model to classify different ethnic groups. | A VGG-Face deep learning model was successfully fine-tuned to classify Malaysian ethnicities (Malay, Chinese, and Indian) based on facial features. |
| 2. Construct a diverse dataset for training and testing the model. | A custom-built facial dataset was collected and preprocessed. Despite being limited in size, it was sufficiently diverse to support meaningful training and evaluation of the classification model. |
| 3. Evaluate the model's accuracy for better recognition of diverse ethnic backgrounds. | The model was evaluated using accuracy, precision, recall, F1-score, and confusion matrices. The results demonstrated robust classification performance enhanced by transfer learning and data augmentation. |

### 6.3 Limitations and Future Work

While this project yielded promising results, several limitations were identified that affect the generalizability, scalability, and ethical implications of the solution. One of the primary limitations was the relatively small size of the dataset. Although data augmentation techniques were employed to increase variation, the limited sample size still constrained the model's ability to generalize across the broader population. Moreover, the dataset may not have fully captured the rich intra-ethnic diversity within each group such as variations in facial structure, skin tone, lighting conditions, and facial expressions which is critical for building a robust and fair model. This limitation becomes particularly important in real-world applications, where image quality and subject presentation can vary widely.

Another concern relates to the nature of the image sources. A large portion of the dataset was sourced from Google, with many images being of celebrities or public figures. These images are typically high-resolution and professionally staged, often involving makeup, filters, or digital enhancements that can distort natural facial features. Such biases may mislead the model during training and impair its ability to perform well on everyday, unedited photographs, limiting its practical applicability in less controlled environments.

In terms of demographic coverage, while this study focused on classifying the three major Malaysian ethnic groups, it does not fully represent the country's broader ethnic landscape, particularly the Bumiputera communities of Sabah and Sarawak, who make up a significant portion of the population. This underrepresentation limits the inclusiveness and equity of the current system. Future work should aim to expand the dataset to include these and other underrepresented groups, thus better reflecting Malaysia's full ethnic diversity. This would involve collecting a larger, demographically balanced dataset through collaborations with local institutions, community engagement, or the use of publicly available datasets, with appropriate ethical consent and privacy safeguards. Incorporating these additional groups would not only improve fairness and representation but also enhance the model's relevance for applications in demographic research, public policy, and inclusive AI systems.

From an ethical standpoint, the task of ethnicity classification using facial features requires careful consideration. Such models can raise concerns around privacy, algorithmic bias, and potential misuse in sensitive areas such as surveillance or profiling. It is essential that future work integrates ethical design principles, including fairness audits, transparency, and the use of bias mitigation techniques to ensure responsible development and deployment.

Additionally, future improvements could include exploring more advanced deep learning architectures, such as EfficientNet or Vision Transformers, which offer better accuracy-to-parameter efficiency. Conducting cross-dataset evaluations would also help in assessing the model’s generalization ability beyond the training distribution. Finally, developing a real-time prototype system could provide practical insights through field testing and user feedback, contributing to both technical and ethical refinement.

### 6.4 Conclusion

This study explored the use of deep learning and transfer learning techniques for facial-based ethnicity classification, with a focus on the three major Malaysian ethnic groups: Malay, Chinese, and Indian. By leveraging the VGGFace model and applying fine-tuning strategies, the models demonstrated substantial improvements in accuracy, precision, and generalization compared to the baseline. The best-performing model achieved over 91% test accuracy, confirming the effectiveness of transfer learning in enhancing classification performance even with limited data.

However, the results also highlighted the challenges and limitations inherent in ethnicity classification tasks. The model’s performance varied across ethnic groups, and was influenced by factors such as data quality, class imbalance, and potential bias from the nature of the image sources. While the fine-tuned models showed minimal overfitting and strong convergence, their reliance on high-quality and celebrity-like images limits applicability in more diverse, real-world scenarios.

Importantly, this work acknowledges the ethical implications of ethnicity classification, emphasizing the need for fairness, inclusivity, and transparency in both dataset construction and model deployment. The current scope does not yet reflect the full ethnic diversity of Malaysia, particularly the underrepresented Bumiputera communities of Sabah and Sarawak. Addressing this gap is crucial for building equitable AI systems that are representative of all societal groups.

## REFERENCES


Anwar, I., & Islam, N. U. (2017). Learned Features are Better for Ethnicity Classification. *Cybernetics and Information Technologies*, *17*(3), 152–164. https://doi.org/10.1515/cait-2017-0036

Al-Shannaq, A., & Elrefaei, L. (2020). Age Estimation Using Specific Domain Transfer Larning. *Jordanian Journal of Computers and Information Technology*, *0*, 1. https://doi.org/10.5455/jjcit.71-1571410322

Ahmed, M. A., Choudhury, R. D., & Kashyap, K. (2020). Race estimation with deep networks. *Journal of King Saud University - Computer and Information Sciences*, *34*(7), 4579–4591. https://doi.org/10.1016/j.jksuci.2020.11.029

Ache, A. I., Kassir, M. M., & Ibrahim, H. M. (2023). Ethnicity Classification: Discriminating between Iranian and Asian populations using hybrid deep learning algorithm. *Journal of Image Processing and Intelligent Remote Sensing*, *41*, 20–33. https://doi.org/10.55529/jipirs.41.20.33

Alghaili, M., AlBdairi, A. J., Li, Z., & Katiyalath, M. (2023). Generating embedding features using deep learning for ethnics recognition. *The International Arab Journal of Information Technology*, *20*(4). https://doi.org/10.34028/iajit/20/4/13

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00444-8

Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, *153*, 46–53. https://doi.org/10.1016/j.compag.2018.08.013

Chandrasekar, A., Clark, S. E., Martin, S., Vanderslott, S., Flores, E. C., Aceituno, D., Barnett, P., Vindrola-Padros, C., & Juan, N. V. S. (2024). Making the most of big qualitative datasets: a living systematic review of analysis methods. *Frontiers in Big Data*, *7*. https://doi.org/10.3389/fdata.2024.1455399

Chicho, B. T., & Sallow, A. B. (2021). A comprehensive survey of deep learning models based on KERAS framework. *Journal of Soft Computing and Data Mining*, *2*(2). https://doi.org/10.30880/jscdm.2021.02.02.005

Cubuk, E. D.., Zoph, Barret., Shlens, Jonathon., & Le, Quoc V. (2019). Randaugment: Practical automated data augmentation with a reduced search space. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 3008-3017. http://doi.org/10.1109/CVPRW50498.2020.00359

Chaki, J., & Dey, N. (2018). *A beginner's guide to image preprocessing techniques*. https://doi.org/10.1201/9780429441134

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. https://api.semanticscholar.org/CorpusID:221112671

Hossin, M., & Sulaiman, MN. (2015). A review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process, 5(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Jahan, S. T., & Mustafa, R. (2024). Ethnicity Detection with Convolutional Neural Network (CNN): Bangladesh Perspectives. *American Journal of Interdisciplinary Research and Innovation*, 5–12. https://doi.org/10.54536/ajiri.v3i4.3772

Jilani, S. K., Ugail, H., Bukar, A. M., & Logan, A. (2019). *On the Ethnic Classification of Pakistani Face using Deep Learning* (Vol. 72, pp. 191–198). https://doi.org/10.1109/cw.2019.00039

Jain, V., Enamorado, T., & Rudin, C. (2022). The Importance of being Ernest, Ekundayo, or Eswari: An Interpretable Machine Learning Approach to Name-Based Ethnicity Classification. *Harvard Data Science Review*. https://doi.org/10.1162/99608f92.db1aba8b

Katti, H., & Arun, S. P. (2019). Are you from North or South India? A hard face-classification task reveals systematic representational differences between humans and machines. *Journal of Vision*, *19*(7), 1. https://doi.org/10.1167/19.7.1

Karkkainen, K., & Joo, J. (2021). *FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation*. https://doi.org/10.1109/wacv48630.2021.00159

Kalkatawi, A., & Saeed, U. (2024). Ethnicity Classification Based on Facial Images using Deep Learning Approach. *International Journal of Advanced Computer Science and Applications*, *15*(2). https://doi.org/10.14569/ijacsa.2024.0150223

Kim, S., Wimmer, H., & Kim, J. (2022). *Analysis of Deep Learning Libraries: Keras, PyTorch, and MXnet* (pp. 54–62). https://doi.org/10.1109/sera54885.2022.9806734

Li, N., Qi, Y., Li, C., & Zhao, Z. (2024). Active Learning for Data Quality Control: a survey. *Journal of Data and Information Quality*, *16*(2), 1–45. https://doi.org/10.1145/3663369

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9992–10002. https://doi.org/10.1109/iccv48922.2021.00986

Maze, B., Adams, J., Duncan, J. A., Kalka, N., Miller, T., Otto, C., Jain, A. K., Niggel, W. T., Anderson, J., Cheney, J., & Grother, P. (2018). *IARPA Janus Benchmark - C: Face Dataset and Protocol*. https://doi.org/10.1109/icb2018.2018.00033

Morales, A., Fierrez, J., Vera-Rodriguez, R., & Tolosana, R. (2021). SensitiveNets: Learning Agnostic Representations with Application to Face Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(6), 2158–2164. https://doi.org/10.1109/tpami.2020.3015420

Mikolajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 117-122. http://doi.org/10.1109/IIPHDW.2018.8388338

Makolo, A. U., & Dada, O. V. (2023). Ethnicity Classification: a Machine Learning approach. *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.7655148

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, *54*(6), 1–35. https://doi.org/10.1145/3457607

Mu, J., Tan, Y., Xie, D., Zhang, F., & Jing, X. (2021). CNN and DCGAN for Spectrum Sensors over Rayleigh Fading Channel. *Wireless Communications and Mobile Computing*, *2021*(1). https://doi.org/10.1155/2021/9970600

Nakada, M., Wang, H., & Terzopoulos, D. (2017). *AcFR: Active Face Recognition Using Convolutional Neural Networks*. https://doi.org/10.1109/cvprw.2017.11

Novac, O., Chirodea, M. C., Novac, C. M., Bizon, N., Oproescu, M., Stan, O. P., & Gordan, C. E. (2022). Analysis of the application efficiency of TensorFlow and PyTorch in convolutional neural network. *Sensors*, *22*(22), 8872. https://doi.org/10.3390/s22228872

Ranjan, R., Patel, V. M., & Chellappa, R. (2017). HyperFace: a deep Multi-Task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(1), 121–135. https://doi.org/10.1109/tpami.2017.2781233

Shin, H., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, *35*(5), 1285–1298. https://doi.org/10.1109/tmi.2016.2528162

Sulfii, M. F. I., Ujir, H., Anak, M. I., & Hipiny, I. (2022). *Malaysian ethnicity classification based on face images using deep learning* (Vol. 9, pp. 73–76). https://doi.org/10.1109/aiic54368.2022.9914030

Vo, T., Nguyen, T., & Le, C. T. (2018). Race recognition using deep convolutional neural networks. *Symmetry*, *10*(11), 564. https://doi.org/10.3390/sym10110564

Wang, W., He, F., & Zhao, Q. (2016). Facial Ethnicity Classification with Deep Convolutional Neural Networks. In *Lecture notes in computer science* (pp. 176–185). https://doi.org/10.1007/978-3-319-46654-5_20

Wang, W., Xie, E., Li, X., Fan, D., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv48922.2021.00061

Wehrli, S., Hertweck, C., Amirian, M., Glüge, S., & Stadelmann, T. (2021). Bias, awareness, and ignorance in deep-learning-based face recognition. *AI And Ethics*, *2*(3), 509–522. https://doi.org/10.1007/s43681-021-00108-6

Yuan, J. (2024). Performance analysis of deep learning algorithms implemented using PyTorch in image recognition. *Procedia Computer Science*, *247*, 61–69. https://doi.org/10.1016/j.procs.2024.10.008

Zhang, Z., Song, Y., & Qi, H. (2017). *Age Progression/Regression by Conditional Adversarial Autoencoder*. https://doi.org/10.1109/cvpr.2017.463

Zhang, Y., & Deng, W. (2020). Class-Balanced training for deep face recognition. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/cvprw50498.2020.00420

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*(3), 107–115. https://doi.org/10.1145/3446776