Face Identification and Face Verification in Spherical Images

MSc Research Project
Data Analytics

Abhilash Rajkumar Kadhane
Student ID: x18203744

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Abhilash Rajkumar Kadhane

Student ID: x18203744

Programme: MSc in Data Analytics **Year:** 2019-20

Module: MSc Research Project

Supervisor: Dr. Catherine Mulwa

Submission Due Date: ………

Project Title: Face Identification and Face Verification in Spherical Images

Word Count: ……………………………………… **Page Count**…………………………………………

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………

Date: ………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□

You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Face Identification and Face Verification in Spherical Images

Abhilash Rajkumar Kadhane
X18203744

Abstract

The advances in omnidirectional vision systems have opened a whole new world of its applications in various domains. This looming technology has increased the popularity and demand of omnidirectional cameras in human-machine interaction systems and surveillance systems. With this rising demand, the requirement of powerful networks to analyse the input spherical images has also increased. Face recognition in spherical images is an unexplored area and should be researched further as it has great potential. Also absence of publicly available spherical face dataset has limited any research in this domain. This project includes face identification implemented using KNN and SVM classifiers and face verification by using Joint Bayesian. Multiple models including a fine-tuned network are implemented for feature extraction and evaluated based on their performance in Face identification and Face verification tasks. Highest accuracy achieved in the Face identification task was 89% (SVM) and 92% (KNN) by the Fine-tuned ResNet-50 model. In face verification task, highest accuracy of 87% (Joint Bayesian) was obtained by the same model. After Fine-tuning an existing pretrained network, remarkable improvement in both the Face identification and Face verification tasks were observed.

Keywords: CNN, Spherical images, Equirectangular projection, Face identification, Face verification

1 Introduction

This section includes the background of the project, motivation for choice of this project, and the gap in the literature it intended to fill. This section also contains the importance of the topic ‘Face recognition in spherical images’ and how it will be beneficial for further research and advancement in this area. Over the past few years, the popularity of omnidirectional cameras has escalated as they provide 360° view of a scene. Images taken by these cameras also known as spherical images, provide more visual information compared to traditional cameras, hence omnidirectional cameras are preferred for many applications. Currently, with the increasing popularity of these cameras in various domains such as autonomous systems (e.g. robots, drones, surveillance cameras), the demand for algorithms and models that can extract meaningful information from spherical images has also increased.

1.1 Project Background and Motivation

Deep learning architectures such as Convolutional Neural Network has shown remarkable performance in image processing as well as computer vision applications like detection, identification, and classification of images (Redmon et al., 2016). However, these architectures fail when the input data is in the form of spherical images. These types of architectures are built to process images represented in the Euclidean space. Thus, spherical images, which are in non-Euclidean space need to be represented in the Euclidean space first so that they can be used as input for these architectures (Lee et al., 2019). This is one of the approaches to address the problem of non-linearity of space where various methods like Equirectangular projection, spherePHD, and cube padding are used to represent Spherical images. But the transformation of spherical image to planar image results in distortions in the image. The shape distortions degrade CNN's performance while extracting meaningful information from the images. In a different approach, spherical image is transformed into multiple perspective planar images, which are then used by CNN for further processing. To tackle the issue of distortion, resampling of spherical image is necessary and to increase the accuracy of the model, large numbers of perspective images are needed. This approach is often denied as huge data is required, which subsequently results in high computational cost.

To fulfil the increasing demand of models and architectures that can understand the inherent nature of spherical images, extensive research has been carried out and is still in progress. Although this research predominantly focused on topics such as object detection and classification in spherical images, there is still scope for research in various other areas of image processing and computer vision. Face recognition is one such field in computer vision. In recent years, significant progress has been done in developing intelligent systems with face recognition techniques. These systems have shown their importance in computer vision domain and various applications such as human-computer interaction, biometrics, and video surveillance (Chen, et al., 2016). Face recognition is often confused with face verification and face identification tasks, however, these two are part of the face recognition problem. Current developments in deep neural networks has demonstrated promising results in both tasks, specifically Convolutional Neural Networks has shown state-of-the-art performance in face recognition problems (Sun et al., 2014). Face recognition tasks have many challenges such as misalignment, high dimensionality in visual data, occlusions, facial expression changes and illumination variations. Despite these obstacles great progress has been achieved in developing of intelligent system that can solve face recognition problems in planar images with the help of available abundant quantity of data. Nonetheless, there is still scope for research regarding the face recognition problem in spherical images.

In summary, the emergence of omnidirectional vision technology gave rise to its numerous applications and introduced new research topics to expand its benefits in various areas like image processing and computer vision. Addressing new challenges regarding the analysis of spherical images have led to development of various spherical networks with promising results in various problems like object detection in spherical images, 3D model recognition. Yet, face recognition in spherical images remains unexplored and requires further research as it is quite beneficial for future of surveillance industry and human-machine interaction systems.

1.2 Project Requirement Specifications

The research question focuses on the comparison and analysis of performances of different face recognition networks on face identification and verification tasks. Models like FaceNet and VGG face network are designed and trained with large planar face dataset to give state-of-the-art results in face recognition tasks. But face recognition in spherical images includes analysis of distorted face images that are obtained after their transformation into input format understandable by the neural networks; these distortions degrade the performance of network. This research also includes creation of large distorted equirectangular version of face image dataset for the training purpose of feature extractors.

***RQ:* "*Can feature extractor models (FaceNet, OpenFace, ResNet-50, and ResNet-50 fine-tuned with spherical face images) for face images enhance/improve face identification and verification in spherical images?".*

Currently traditional feature extractors fail to get exact features from spherical images; hence the need of feature extractors which can analyse these images. Face recognition in spherical images is difficult due to distorted face images. Therefore, analysis of spherical faces images is important and requires usage of techniques different from traditional ones.

Sub-RQ: "*To what extent can face recognition models, with or without fine-tuning on omnidirectional face images data (generated equirectangular version of face image dataset), be used successfully to perform face identification (KNN, SVM) and face verification tasks (Joint Bayesian) on spherical images?"*

This is significant and will benefit practitioners and stakeholders in the surveillance industry. The contributions resulting from developed solutions are important for introducing new and efficient techniques for surveillance systems and human-machine interaction systems.

1.3 Research Objectives and Contribution

To investigate the research question and sub research question, a series of research objectives and sub-objectives were defined and implemented as shown in the Table 1.

Table 1: Summary overview and description of research objectives

ID	Name	Description	Evaluation
1	Literature Review	Identify and critically review the literature on spherical image processing and face recognition.	
2	Prepare VGGFace2 data for dataset generation	Normalize face images and get landmarks of each image to prepare dataset for transformation.	
3	Generate dataset	Create equirectangular version of curated VGGFace2 dataset.	
4	Data pre-processing	Prepare dataset for training neural networks. Normalize face images, perform other required face image pre-processing steps.	

5	Implementation of data modelling	Implementation of face recognition models using machine learning techniques to extract features from face images	
5.1	OpenFace	Implement OpenFace model for feature extraction in spherical images data. Evaluate and Result with the help of classifiers.	Accuracy, Precision, Recall, F-score
5.2	ResNet-50	Implement ResNet-50 model for feature extraction in spherical images data. Evaluate and Result with the help of classifiers.	
5.3	FaceNet	Implement FaceNet model for feature extraction in spherical images data. Evaluate and Result with the help of classifiers.	
5.4	ResNet-50 fine-tuned with generated data	Implement ResNet-50 model, fine-tuned with generated data, for feature extraction in spherical images data. Evaluate and Result with the help of classifiers.	
5.5	Classification model for face identification	Implement SVM and KNN classifier models for face identification task.	
5.6	Binary classification model for face verification	Implement Joint Bayesian binary classifier for face verification task.	
6	Data visualization	Display identity clusters of face images with the help of t-SNE.	
7	Compare results	Compare developed models with their performance in face identification and face verification tasks.	

The contribution of this project includes creation of equirectangular projection of face images (VGGFace2) dataset; implementation of face identification models with the help of feature extracting neural network and SVM and KNN classifier; implementation of face verification models with the help of feature extracting neural network and Joint Bayesian binary classifier; performance comparison of various feature extracting networks.

The remaining report is structured as follows: Section 2 includes review of the current literature surrounding spherical images (their representations and analysis) and face recognition, Section 3 is methodology and techniques used, Section 4 is implementation, evaluation and results section followed by Section 5 includes discussion chapter and finally Section 6 the conclusion and future work.

2 Literature Review

2.1 Introduction

In this section, literature review on the topic and few related key areas is presented in the following of sub-sections. As the topic of this project is face recognition in spherical images, literature has been reviewed covering areas: spherical images, image processing in spherical images and face recognition. (2.2) sub-section includes detailed discussion on spherical images and their representations. (2.3) sub-section includes review of literature of various machine learning models and techniques used for spherical image analysis. (2.4) sub-section discusses face recognition models and architectures.

2.2 Literature Review on Spherical Images and Representations

Although spherical images have many benefits, processing and analysing these images with conventional deep learning methods is a task full of challenges. One of these challenges includes conversion of spherical image signals into the signals that model can understand, This conversion is merely the transformation of spherical image signals defined on a sphere to that on a regular planar grid structure, as regular planar grids are unavailable due to the nonlinearity of the spherical space. Two basic approaches are used in this case: 1) Converting spherical image into multiple perspective planar images and feeding these images to deep learning CNNs for further processing. This approach solves the distortion issue caused in planar image during the transformation from spherical to regular planar grid. But it requires re-sampling the spherical image and, perspective images are required in large quantity for high accuracy. This requirement of large data increases the computational cost, which is undesirable. 2) Spherical images mapped into different representations such as ERP i.e. equirectangular projections. Transforming spherical images into these representations result in loss of continuity along with shape distortions because of nonuniform spatial resolving power.

Transformation of spherical image from non-Euclidean space to Euclidean space results in loss of its cylindrical properties and continuity. In object detection tasks single objects are often misunderstood as two different objects due to discontinuity along the borders of ERP image. Monroy et al. (2018) proposed cube map representation to address the discontinuity issue caused during transformation. It was observed that the proposed representation gets less affected by rotation as compared to ERP. Although the proposed representation reduces the variance of spatial resolving power, variance to the edge of the cube face from their centres remains. Furthermore, as bottom and top sides of the cube map are orthogonal to other sides, defining kernel orientation to extract their uniform locality information is ambiguous. Lai et al. (2018) tackled the drawbacks of ERP representation images by proposing usage of normal field of view images. Equirectangular representations of omnidirectional videos have considerable amount of distortions and often only few parts of the scene in the space and time are equally important. This condition encouraged the research to be focused on scene semantics and visual saliency. Approach included creation of saliency maps from omni-directional images. After creating maps, NFoV images were extracted particularly from high saliency areas. This approach resolved distortion issue and achieved high accuracy with the help of NFoV image sampling, however it was not able to utilize omnidirectional view completely for maximum visual information.

Lee et al. (2019) proposed a method which utilizes spherical polyhedron representation of spherical images. New convolution and pooling techniques were introduced for the proposed representation. Spherical polyhedron representation was obtained from an icosahedron, which makes it less irregular compared to other representations including the equirectangular representation and the cube map representation. This representation also performed well due to minimized variance of spatial resolving power of the sphere surface. For the demonstration of feasibility of the proposed approach; detection, segmentation and classification tasks were performed on common datasets MNIST, Standford2D3D and SYNTHIA. Spherical polyhedron representation managed to overcome some of the drawbacks that were present in Equirectangular representation. Although there were several issues with usage of equirectangular representation, such as discontinuity at the borders, distortion effects and rotational constraints, many researchers have chosen the ERP-based methods because of its simplicity and tried resolving those issues with different techniques.

Coors et al. (2018) proposed a method to solve the issue of nonuniform resolving power. Invariance against the distortions was handled by proposed deep learning network by encoding them into CNN. The approach used in this research included sampling of pixels with rate proportionate to latitude from Equirectangular projection image. The proposed model wraps the convolutional filters around the sphere, subsequently adapting the sampling locations to effectively reverse the distortions. Effectiveness of the proposed network was demonstrated by performing object detection task on real-world omnidirectional datasets.

2.3 A Critique of Methods and Techniques for Analysing Spherical Images

Recent research in deep learning techniques has made great progress in computer vision domain. Especially, Convolutional Neural Networks has shown promising results in the field of image processing and analysis. Despite the cutting-edge performance of the CNN in image and video analysis tasks, these networks failed to deliver exceptional results while analysing the omnidirectional images. The underlying projection models of the traditional CNNs differ from that of Omnidirectional image data, hence the degraded performance. A few recently developed techniques managed to solve this problem by projecting spherical image on a planar surface, however it affected the accuracy and efficiency of model. Some of these techniques tried retaining the accuracy by repeatedly projecting spherical image on tangent planes but suffered high computational cost. A learning-based solution was proposed by Yu-Chuan et al. (2018) to keep both efficiency and accuracy. The proposed method was able to maintain the model efficiency, as it was learning CNN which processes spherical image into its equirectangular representation. The Proposed model also retained the accuracy by mimicking the flat filter responses produced on all tangent plane projections by existing network. Proposed method handled distortions in image by adjusting the network structure and following a systematic procedure. The training process was accelerated by a proposed kernel-wise pre-training procedure. In this research, successful transfer of FOV 2D imagery domain models into spherical image domain was demonstrated using PASCAL and Pano2Vid datasets.

Many research studies followed a method of partitioning spherical image into multiple sub images to produce NFoV images. Even though this approach can be used to deal with distortion issue, extraction of NFoV image is not as rewarding as utilization of complete omnidirectional image view. A distortion-aware convolutional network was proposed by Qiang et al. (2018) to resolve distortion issue in images. In this network, non-regular grid is sampled for each pixel depending on the distortion level, and convolution is performed on it with the help of square kernels shared by all pixels. The issue of discontinuity was also fixed by this approach. The proposed network was tested on transformed MNIST and CIFAR-10 datasets and showed better results than the conventional CNNs. A similar approach was used by Tateno et al. (2018) to handle distortions while estimating the panoramic depth from a single image. Learning based approach was used to train on RGB images and revert it to omnidirectional panoramic images. The distortion-aware convolutional filters used in this method were able to modify their receptive field by deforming the shape as per the distortion and projection model.

Khasanova et al. (2017) performed image classification task by considering geometry of 360° cameras with graph-based representations. In this research deep neural networks were extended to data on graph. This method succeeded in constructing graph so that the convolutional filters works in same way for similar pattern on different positions of image even if distortions existed. The proposed method included few assumptions with respect to mapping projection, and therefore its implementation was restricted to specific projections. Although there are some restrictions with projection mapping, proposed graph-based neural network was evaluated on modified datasets MNIST-012 and ETH-80 and worked reasonably well.

Patterns on planar image move around causing movement in the form of translation. Whereas patterns on spherical images move around with movement in 3D rotation. A spherical CNN named S2-CNN was developed by Cohen et al. (2018) to detect patterns on the sphere regardless of their rotation. In this research a generalized Fourier Transform algorithm was used to define rotation-equivariant and expressive spherical correlation, as it was observed that spherical correlation fulfils the generalized Fourier theorem. To evaluate the performance of the proposed method an experiment was performed on MNIST dataset and its rotated version. Planar CNN (a model using traditional convolutional layers) was used as a baseline method for comparison purpose. In this experiment, combinations of MNIST dataset (non-rotated) and its rotated version were utilized for training and testing. It was observed that the accuracy achieved by traditional convolutional network was higher than that of the proposed spherical CNN if non-rotated version of dataset is used for training and testing purpose. However, utilizing rotated version of dataset resulted in lower accuracy in planar CNN as compared to spherical CNN.

2.4 A Critique of Face Recognition Methods and Techniques and Identified Gaps

In the last few decades there has been a great demand of intelligent systems with efficient face recognition techniques to analyse images including human faces. With the emergence of deep learning methods and convolution neural networks great progress has been made in the domain of face identification in wild (Wan, et al., 2017). Based on the usage of CNN, an extensive evaluation of face recognition system was conducted by Yang et al. (2015). In this research, three different CNN architectures were implemented which were differed by number of convolutional layer and fully connected layers. For face recognition task a Joint Bayesian model was used as a metric learning method. This research introduced several important properties of the CNN Face Recognition System e.g. dimensionality of the feature vectors that can be reduced without hampering the accuracy of the model. After comparing different CNN architectures based on their implementation choices and quantitative evaluation impact on face recognition, it was observed that few existing architectures such as Fisher Vector Faces gave impressive results. Parkhi et al. (2013) proposed a novel approach for face verification in LFW face dataset, which included a new image representation based on dense SIFT feature and Fisher vector encoding. This approach gave promising results along with some newly improved techniques. These techniques included learning of compact descriptor from high dimensionality Fisher vectors with the help of discriminative metric learning. In this research, a framework was also introduced to reduce large margin dimensionality. Although this research majorly worked on high dimensionality vector problems and gave important findings related to it, there was a limitation in method’s application as Fisher vectors reduction system is based on single feature type.

Face recognition basically works on a principle of reducing intra-personal differences while increasing the inter-personal differences. Sun et al. (2014) proposed a solution, which includes deep learning method with face verification and identification signals as learning parameters. The approach used in this research, focused on drawing feature vectors of different identities far from each other to increase the inter-personal variations, and pulling them together if they belong to same identity so that intra-personal distance will be reduced. Applying this approach on face verification in labelled faces in the wild dataset achieved 99.15% accuracy.

Many supervised learning models gave promising results in face identification problem under an unconstrained environment. But creation of labelled dataset in large scale is time consuming as well as costly task. To avoid this situation, semi-supervised learning method was used by few approaches. Lin et al. (2017) demonstrated an experiment to get improved results by combining active learning and self-paced learning technologies. In this framework, a batch of classifiers was progressively maintained with increasing face images of individuals. Before performing this step, classifiers were initialized with few annotated individual samples. For this initialization process, feature vectors of images were extracted using Convolutional Neural Networks. To update the classifier, the next step included selection of candidates from unannotated samples and their ranking based on prediction confidence. This approach utilized high-confidence samples in self-paced way and low-confidence samples in user-query way. Later neural network was tuned based on updated classifiers.

Face recognition task is nothing but re-identification, where re-identification system gives an output in the form of similarity score between two input images. Input pair of images are classified in re-identification system whether they are similar or not depending on condition if images depicts the same person or different people. A learning-based approach was proposed by Shen et al. (2015) for person re-identification problem, particularly based on low-level hand-crafted and high-level visual features. This evaluation measure is very common in person re-identification evaluation technique and in this research, it was optimized using formulated optimization algorithms. Face recognition model developed by Schroff et al. (2015) used triplet loss function and trained on 100 million to 200 million labelled images. A large-scale experimentation carried out in the research led it to achieve state-of-the-art performance. Similarly Amos et al. (2016) developed a variant of the same model with triplet loss function and managed to deliver good results even after being trained on very small dataset.

2.5 Conclusion

The above literature review gives a brief idea of the research done in spherical image analysis and face recognition domain. From this review, to the best of candidate’s knowledge, not enough research has been accomplished in the field of face recognition in spherical images. However, most of the research done in spherical images domain was focused on object detection, cosmological applications, and molecular 3D model recognition. There are many powerful networks (e.g. OpenFace network, FaceNet network) available for face image analysis which have given remarkable performances in face recognition in planar images but never tested on spherical images. Additionally, to achieve promising results the face recognition model requires large face image dataset for its training purpose, and there is no such face dataset (in spherical images) available publicly to promote the research.

3 Face Recognition Methodology and Technique Used

3.1 Introduction

This section represents the methodology approach and techniques used in this project. This project covers both face identification and verification tasks, with goal of research on performance of different architectures while analysing spherical face images.

3.2 Face Recognition Methodology Approach

The motivation of the research is to perform face recognition task in spherical images and subsequently face identification and verification tasks on same spherical face images dataset with the same set of feature extractors. After the successful feature extraction different approaches were designed for each task. Figure 1 shows complete workflow of proposed approach for both face verification and face identification tasks. As discussed, generated data and feature extractors are common for both tasks and this step in the workflow (till the training of model) is performed only once. Later face identification is performed using SVM and KNN classifiers and face verification with the help of Joint Bayesian binary classifier. Following sub-sections give stepwise information about the designed workflow approach.

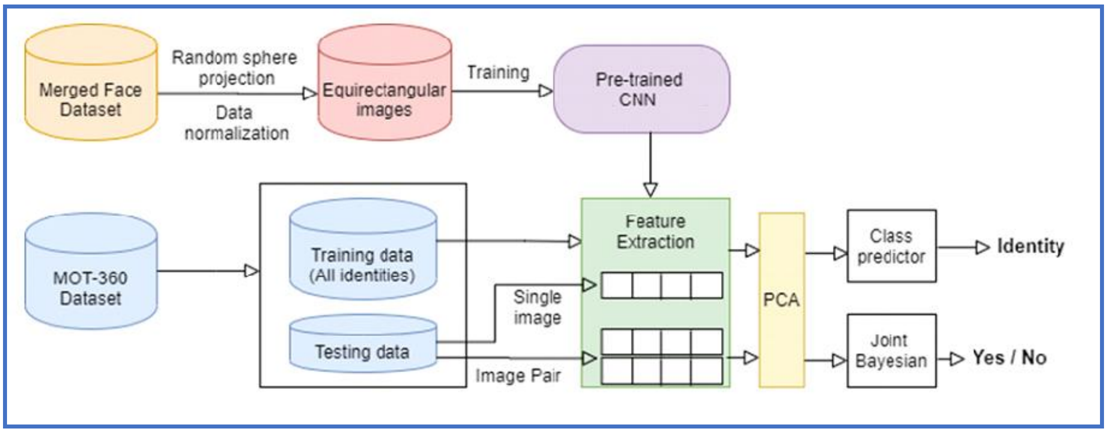


Figure 1: Workflow for face identification and face verification.

3.2.1 Data Selection

The performance of the image processing models improves with the quantity of the data used for its training. Similarly feature extracting models in a face recognition system requires large face image data for training purpose. VGGFace2[1] dataset, which has more than 9,000 identities and 3.3 million images, was selected for fine-tuning of feature extractor. All face images in this dataset are captured ‘in the wild’, with variations in emotion, pose, different occlusion and lighting condition. This dataset was available in planar form and for this research it was transformed into its equirectangular version. MOT-360 dataset produced by Cirne et al. (2019) has annotation details publicly available along with the dataset. These annotation details include eye centres and camera angles, which were utilized during generation of equirectangular version of the VGGFace2 dataset. Further MOT-360 dataset was used to train the metric learning system and test the face verification and face identification methods. MOT-360 dataset comprises a total of 7,409 equirectangular representations of spherical face images. A total of 52 identities are included in this dataset with minimum 25 images per identity.

3.2.2 Preparing VGGFace2 for Dataset generation

Before using the VGGFace2 face data to generate spherical face image dataset (i.e. equirectangular version of VGGFace2, required for research) it was cleaned and normalized with the help of Dlib and OpenCV python libraries. Many images in the VGGFace2 dataset have more than one face in a single image, detection of the appropriate face and capturing

[1] http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/

required landmarks is done with help of python scripts. The images in which face cannot be detected were removed from the dataset. A metadata was created containing Image details and landmarks of the face detected in it.

3.2.3 Generate Equirectangular Images

This research is focused on spherical image processing and with the motivation of comparing performances of various architectures analysing distorted images (distortion caused by transformation from non-Euclidean space to Euclidean space). This distortion effect was simulated in VGGFace2 face images by projecting each image on a sphere and then obtaining its equirectangular representation. This principle was used for generating equirectangular version of VGGFace2 data. With the help of camera angles and landmark details provided by MOT-360 dataset and the scripts in C and Python equirectangular version of VGGFace2 was generated. Figure 2 shows the projection resulting from transformation of spherical grid into the planar grid. This gives an idea of the level of distortions at every position on sphere including polar regions as well as equatorial region. Distortions in the face image projected on sphere is seen to be gradually increasing as it moves towards polar regions from equatorial region.

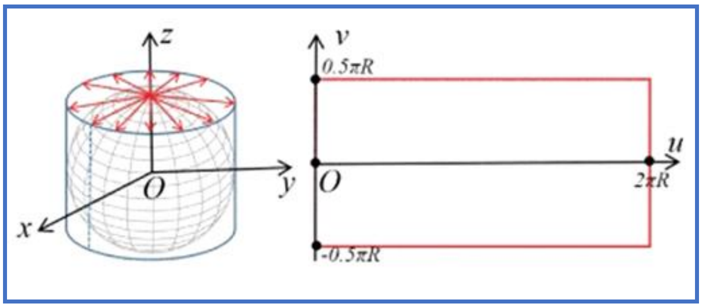


Figure 2: Spherical projection to Equirectangular projection.

3.2.4 Data Pre-processing

Performance of the image processing model improves with the quantity of the data used for its training, but most importantly bad samples and impurities in the dataset must be removed. Face image data pre-processing used in this project includes few important steps as follows:

- Normalization: With the help of OpenCV2 library perform visual normalization to fix very light/dark pictures.
- Face and eyes detection: Face and landmark detection is done with help of Dlib python library and predictor ‘shape_predictor_68_face_landmarks.dat’.
- Face cropping: For better results from face recognition models, removing external distracting information by cropping faces with help of Dlib and OpenCV2 libraries.
- Image resizing: Resizing face images as per the required input dimensions of the feature extractors.
- Class imbalance: Unbalanced classes in training dataset can be a serious problem, and it is difficult to split dataset into training and validation in such cases. Therefore, removing class imbalance.

Face dataset MOT-360 and equirectangular version of VGGFace2 data were both pre-processed using above mentioned steps and utilized by feature extractors and fine-tuned model respectively.

3.2.5 Training Network for Feature Extraction

A generated equirectangular version of VGGFace2 dataset was used to train and validate the feature extractor. Training neural networks with large datasets requires a lot of time as well as resources. To fulfil the requirement of high computational power and resources and utilize free of cost TPU accelerator and Tesla p100 GPU accelerator availed by Google Colaboratory, feature extracting neural networks were trained on Google colab platform. It was observed in an experiment that TPU takes more training time than GPU when batch sizes are small. However, on increasing the batch size TPU gets faster and outperforms GPU[2]. Due to some restrictions in Colab, dataset was loaded and utilized in batches for training of feature extractor. After completion of fine-tuning the feature extractor, weights of the fine-tuned model were saved and utilized at the time of feature extraction in face recognition tasks.

3.2.6 Feature Extraction

For feature extraction different CNN models were implemented. These models were either pretrained models or fine-tuned with generated data. MOT-360 data was pre-processed and fed to these models to obtain feature vectors for each of the face images in dataset. Face images were resized as per the input dimension requirements of the models. After completion of feature extraction, all features were split into training and testing sets for face identification and face recognition tasks. Feature extraction is presented in chapter 4

3.2.7 Face Identification

Feature vectors obtained from feature extracting neural networks, were forwarded to SVM and KNN classifier to predict the identity class. In other words, face images were classified into one of n (e.g., n = 52) different identities using these classifiers. KNN and SVM classifiers work on different principles and have their own benefits of usage, therefore both were used for the classification. KNN is automatically non-linear. It can detect non-linear and linear distributed data and classify as per the nearest neighbors of data point. Also it performs good with a lot of data points. Whereas SVM try to find a hyper-plane separating the classes in training data. Therefore, it was important to consider both classification techniques. MOT-360 dataset was used for training and testing of the classifier. As MOT-360 dataset has only 52 identities, all these identities were used to train the classifier. The dataset is divided into 80% for training and 20% for testing.

3.2.8 Face Verification

Classifier construction has been widely studied as a key component of face verification system (Lu and Tang, 2014). Face verification task includes classification of input image pairs into two classes: images belong to same identity or images belong to different identities. A binary classifier was used here to get the probability of two face images if depict the same identity. Joint Bayesian has shown state-of-the-art results in face verification (Cao, 2013). In this method, extracted facial features (x) are represented as a summation of two independent Gaussian variables as shown in below formula.

[2] https://colab.research.google.com/github/zaidalyafeai/Notebooks/blob/master/GPUvsTPU.ipynb

$x = \mu + \varepsilon$,

where $\mu \sim N(0, S\mu)$ represents the face identity and $\varepsilon \sim N(0, S\varepsilon)$ is intra-personal variations. P(x1, x2 | HI) and P(x1, x2 | HE) are the intra- or extra-personal variation hypothesis given by Joint Bayesian model’s joint probability of input faces. These two probabilities are also Gaussian with variations, as per the above equation. While testing, the likelihood ratio is calculated from these probabilities as shown below.

$r(x1, x2) = \log (P(x1, x2 \mid HI) / P(x1, x2 \mid HE))$

It has closed-form solutions and it is seen to be more efficient than the other techniques. Classification of face pair is done by comparing this likelihood ratio to a threshold optimized in training process. MOT-360 dataset was divided into 70% training and 30% testing sets, for training and testing it on Joint Bayesian classifier. 30% of portion dataset was chosen only to get 6,000 positive pairs (same person images) and 6,000 negative pairs (different person images).

4 Implementation, Evaluation and Results of Face Recognition Models

4.1 Introduction

This section presents the implementation of the ICT solution for face identification and face verification in spherical images, to support practitioners and stakeholders in surveillance industry with new omnidirectional vision systems. It includes detailed explanation on the choice of networks for feature extraction, and technologies for image processing, data generation and pre-processing. Following are the details on implementation and evaluation of the ICT solution.

Implementations: Section 4.2 includes detailed information about methods and techniques used to generate the equirectangular version of VGGFace2 dataset. This dataset was later used for fine-tuning ResNet-50 network. Further sections include details about implementation of ICT solution for face recognition problem using combinations of OpenFace model (section 4.3), ResNet-50 (section 4.4), FaceNet model (section 4.5), Fine-tuned ResNet-50 model (section 4.6) as feature extractors and KNN/SVM classifiers for face identification, covering objective 5.5. Similarly all the feature extractors mentioned above were combined with Joint Bayesian classifier for face verification solution, which covers objective 5.6.

Evaluations: To evaluate and validate the implemented feature extractors and classifiers (used for face recognition) two experiments were conducted: 1) Face identification (performed using KNN and SVM classifiers on features extracted by implemented feature extractors) 2) Face verification (performed using Joint Bayesian binary classifier). Herlocker and McLaughlin (2017) stated that machine learning algorithms can be evaluated using several evaluation functions. The following evaluation methods and metrics were used in this project: Accuracy, Precision, Recall, F score. Assuming TP is true positive, FP is false positive, TN is true negative, and FN is false negative. Then following formula calculates accuracy which is nothing but closeness of derived value to the standard known value (Lantz, 2013).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

On the other hand, Precision is defined as ratio of correctly predicted positive values to the total predicted positive values. If the model has high precision then it can be said that it has high chances to be true (Lantz, 2013). This evaluation metric was important in face verification to find the if the model correctly depicts same/different identity of the input face images. Following formula is used to calculate the precision.

$$Precision = \frac{TP}{TP + FP}$$

Similarly, Recall is ratio of correctly predicted positive values to all values in actual class. If the model has high recall it interprets that it has low false negatives. Following formula is used to calculate the recall.

$$Recall = \frac{TP}{TP + FN}$$

For F1 score calculation both Precision and Recall both evaluation metrics are used. It is the weighted average of Precision and Recall. F1 score is included in this evaluation method due to presence of uneven class distribution. Following formula is used to calculate the F1 score.

$$F1Score = \frac{2 * Recall * Precision}{Precision + Recall}$$

4.2 Creation of Equirectangular Version of VGGFace2 Dataset

This section covers objective 2 and objective 3, which are preparing VGGFace2 face dataset for dataset generation and generating equirectangular version of VGGFace2. VGGFace2 dataset, publicly available, contains images from around 9,000 identities spanning a wide range of different professions, ethnicities, and ages. These images are downloaded from Google Image Search and have large variations in illumination and pose. Figure 3 gives exploratory details of the dataset. Additionally, the face image distribution for different identities varies from minimum 87 to maximum 843, with an average of 362 images per identity. This large-scale face dataset was transformed into equirectangular version and further used to train feature extractor for better results.

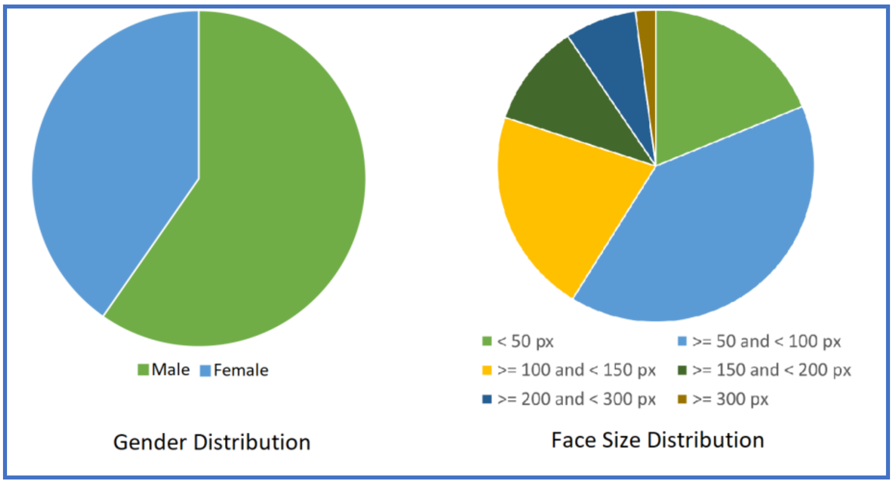


Figure 3: VGGFace2 dataset details

4.2.1 Prepare VGGFace2 Data for Dataset Generation

Distortions in images can be simulated by projecting the image on a random position in a sphere, with this principle Cirne et al. (2019) proposed a technique to generate spherical images from its planar form. To generate equirectangular version of VGGFace2 dataset, the same technique was used in this project. This technique requires landmark details of the face in the image, particularly left eye and right eye landmarks. Hence it was necessary for each image to be correctly annotated. To prepare VGGFace2 data for equirectangular version dataset generation, landmarks (left eye and right eye) were detected using front face detection functionality of Dlib library and 'shape_predictor_68_face_landmarks.dat'. In many of the images more than one faces were captured. By detecting landmarks of the most accurate face in the image and discarding other face landmarks, script made sure that only one face gets selected. Also in this process face image was discarded if the predictor failed to detect the eyes. This step helped to create metadata of all the face images consolidated with their file path, label, and landmark details. This metadata was used in the next step of dataset generation.

4.2.2 Generation of Equirectangular Version of VGGFace2 Dataset

Creation of equirectangular version of face image was done in 2 steps; first is projecting planar face image on sphere and second is obtaining its equirectangular representation. In the first step, cropped image section containing face portion, detected with the help of landmarks, was projected on sphere using randomly chosen camera angles provided by Cirne et al. (2019). This transformation is performed with the help of python library skimage and its transform functionality. Transformation caused by projection of planar image on the sphere was captured in the Portable Pixmap format temporary file and later used for the next step. The obtained temporary file was further processed to get the equirectangular representation of the projected face image. Figure 4 shows the distortion achieved in face images in the equirectangular version of VGGFace2 dataset. This dataset was further pre-processed and used to train and validate the feature extractor ResNet-50, i.e. fine-tuning of the feature extracting networks.

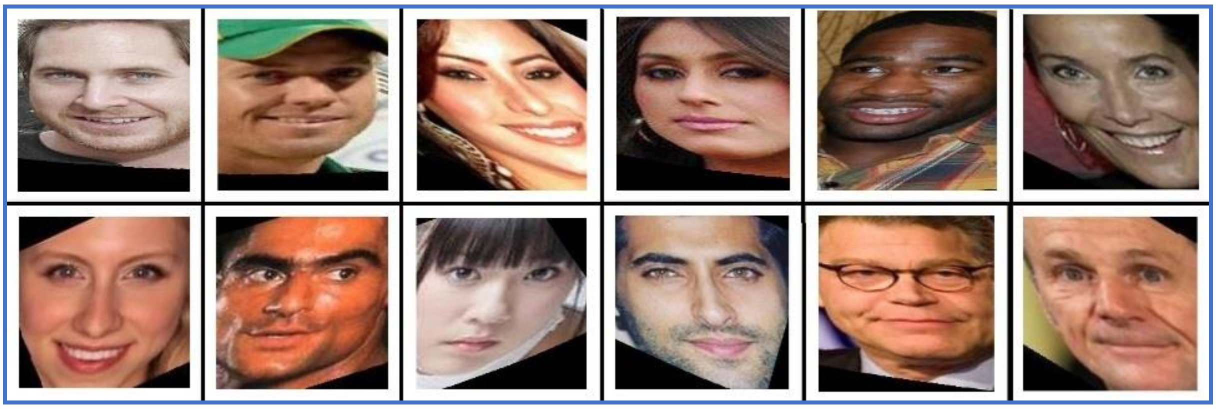


Figure 4: Equirectangular version of VGGFace2 dataset

4.3 Implementation, Evaluation and Results of OpenFace Face Recognition Model

This section includes OpenFace model implementation for face identification and face verification tasks, this covers objective 5.1. OpenFace was chosen, as per the literature, as proposed by Amos et al. (2016) model showed that competitive accuracy and promising results on the LFW dataset with very small training dataset. Model used for the implementation of face identification and verification system is variant of NH4 architecture designed by Schroff et al. (2015). This network has fully connected layer with 128 hidden units followed by an L2 normalization layer on top of the convolutional base. These layers are also known as embedding layers, which give 128-dimensional embedding vector.

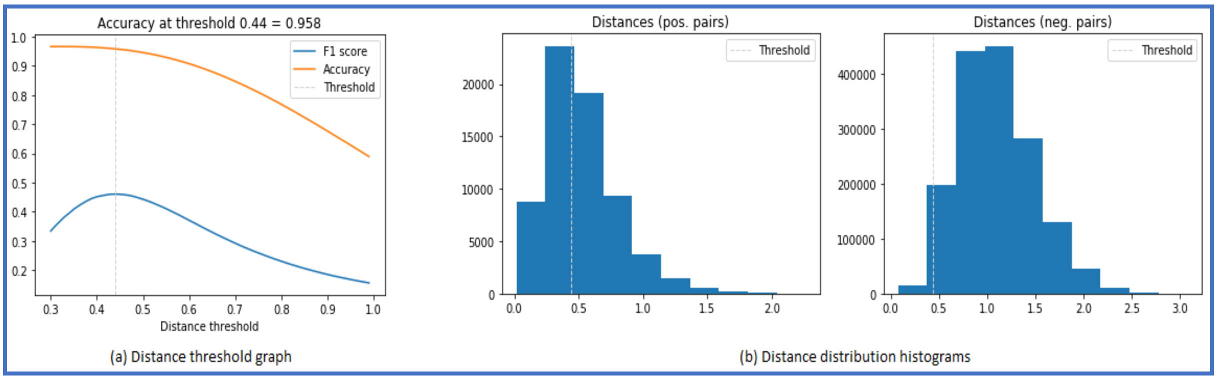
4.3.1 Implementation of OpenFace Feature Extractor

OpenFace network, used for feature extraction of face images in MOT-360 dataset, is pre-trained with public face recognition datasets CASIA-WebFace and FaceScrub datasets. Keras was used for the implementation of CNN, and after creating the model it was loaded with weights provided in the binary format 'small2.v1.h5'. Before feature extraction step dataset was pre-processed as per the steps mentioned in the methodology/Data pre-processing section. Model requires input image in (96, 96, 3) dimension, hence data pre-processing also included a step to prepare dataset for model (such as resizing into appropriate shape). This process was important as it ensures that faces are aligned before feeding them to the model. With the help of implemented model the embedding vector, 128-dimenstional representation, were extracted for face from aligned input image.

Using OpenFace network feature extraction on face image dataset was performed. Next section includes analysis of the extracted features (section 4.3.2).

4.3.2 Results of Extracted Features

The obtained embedding vectors of input images were analysed for performance of the feature extractor. Model works on the Euclidean distance measures for its decision of same/different identity of the embedding vector. The squared L2 distance between same identity image vectors must be smaller than that of different. For the given dataset, optimal value of this distance (also known as τ) was calculated and compared on the basis of F1 score as evaluation metric. Figure 5 shows (a) the graph plotted of distance threshold against accuracy and F1 score, and (b) histogram showing distributions of positive and negative pairs with the location of decision boundary.

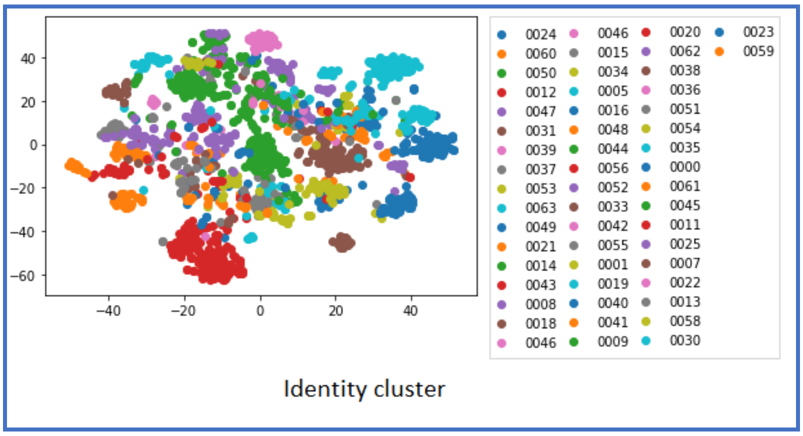


(a) Distance threshold graph

(b) Distance distribution histograms

Figure 5: Analysis of embedding vectors

Dataset used in this implementation had total 52 identities i.e. 52 classes. For data visualization, embedding vectors of these images were plotted on a 2D space to visualize the identity clusters. Objective 6 was covered by visualizing the feature vectors for identity clusters. With the help of t-distributed Stochastic Neighbor Embedding technique 128-dimensional embedding vectors were plotted on a 2D graph as shown in Figure 6. In this figure each colour represents an identity and each coloured dot on the graph represents embedding vector of the image.



Identity cluster

Figure 6: Identity cluster using t-SNE

Extracted features were used for the face identification with the help of SVM and KNN classifiers. These classifiers were implemented and evaluated in next section (section 4.3.3).

4.3.3 Implementation, Evaluation and Results of SVM and KNN Classifier on Features Extracted by OpenFace

Face Identification task was implemented with the help of SVM[3] and KNN[4] classifiers, by training them on labelled embedding vectors obtained from feature extraction. These classifiers work as a database, in context of face recognition, used for predicting labels/identities of input face images. In KNN, also known as k-Nearest Neighbors algorithm, classification is performed on the basis of plurality vote of the neighbors of the object, and the class most common among its k nearest neighbors is assigned to the input. On the other hand, Support Vector Machine constructs a hyperplane or set of hyperplanes in high-dimensional space for classification. These classifiers were implemented using sklearn library in python. For Face identification task, embedding vectors extracted using implemented network were divided into 80% for training and 20% for testing.

KNN achieved 89% accuracy on the test dataset, whereas SVM achieved 85% accuracy. Identity cluster shown in Figure 6 shows existence of multiple small clusters formed for each identity resulting higher accuracy in KNN as compared to accuracy achieved in SVM.

4.3.4 Implementation, Evaluation and Results of Joint Bayesian Classifier on Features Extracted by OpenFace

Face verification task is implemented with the help of Joint Bayesian Face Verification algorithm[5]. As discussed in the methodology section, facial feature is represented in terms of two independent Gaussian variables: ε for intra-personal variations and μ for the face identity. These two obey Gaussian distributions: N(0, Sε) and N(0, Sμ), and the log likelihood ratio r (x1,x2) is calculated using two covariance matrix Sε and Sμ. The approach used in verification tasks follows Expectation-Maximization algorithm to find these covariance matrices between two classes. At the end, obtained log likelihood ratio is compared with the threshold such that, if ratio >= threshold, it is considered to be the same person otherwise different.

Dataset was divided into 70% for training and 30% for testing. Reason for using 30% testing data was to generate sufficient positive and negative pairs of images (as 20% testing data was unable to produce more than 4,000 positive pairs). Random 6,000 positive pairs (i.e. both images in the pair belongs to same identity) and 6,000 negative pairs (i.e. images in the pair belongs to different identities) were generated programmatically. After training the Joint Bayesian classifier with training set, pair list containing all positive and negative pairs were fed to the classifier along with test dataset. Likelihood ratio obtained for each of the pair was compared with threshold to predict if pair belongs to same identity or not. Later these predictions were used to generate the classification results with the help of sklearn.metrics library. Joint Bayesian achieved 84% F1 score for face verification task using feature extractor based on OpenFace.

The presented results have solved the research question (Section 1, sub-section 1.2) covering implementation as well as the, evaluation of one of the mentioned feature extractors with the help of KNN/SVM and Joint Bayesian classifiers.

[3] https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

[4] https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26

[5] https://blog.csdn.net/cyh_24/article/details/49059475

4.4 Implementation, Evaluation and Results of ResNet-50 Face Recognition Model

This section has implementation of ResNet-50 model for face recognition tasks and this covers objective 5.2. ResNet-50, deep learning convolutional neural network with 50 layers, has shown state-of-the-art performance in standard face recognition. In this network K. He et al. (2016) have reformulated layers as learning residual functions with reference to inputs of layer, not the learning unreferenced functions. It was shown that these residual networks are easy to optimize and gets accuracy from increased depth. Network implemented in this section was imported from keras-vggface library[6] and its installation process is included in the configuration manual.

4.4.1 Implementation of ResNet-50 Feature Extractor

ResNet-50 was used for the feature extraction of the face images in MOT-360 dataset. On the first occasion creating the model, library downloads the model weights and loads into the model. In this implementation pre-trained ResNet-50 model was used, and model was trained on 8,631 identities in the MS-celeb-1M[7] dataset. As ResNet-50 expects input image in (224, 224, 3) shape, required step was added in pre-processing of the images in dataset. Implemented ResNet-50 network returns embedding vector for input image, to eliminate unnecessary noise in the feature vector and dimensionality reduction PCA was implemented. But it was removed later due to degraded performance in face identification task.

After extracting feature using this model, they were further used for the face identification with the help of KNN and SVM classifiers (implemented in the next section 4.4.2); And for face verification with Joint Bayesian classifier (section 4.4.3)

4.4.2 Implementation, Evaluation and Results of SVM and KNN Classifier on Features Extracted by ResNet-50

Features extracted using ResNet-50 network were divided into 80% for training set and 20% for testing set. SVM and KNN classifiers were used for face identification task. These classifiers were trained with the training set of feature vectors and their labels. KNN achieved 79% accuracy on the test dataset, whereas SVM achieved 63% accuracy.

4.4.3 Implementation, Evaluation and Results of Joint Bayesian Classifier on Features Extracted by ResNet-50

Face verification task, on features extracted using ResNet-50 network, was performed with the same approach followed in face verification task with OpenFace network implementation. Joint Bayesian classifier implemented for this task was trained with 70% of training data. 30% testing data was chosen in order to create 6,000 positive pairs (i.e. face images of same identity) and 6,000 negative pairs (i.e. face images of different identities). These pairs were fed to trained Joint Bayesian classifier along with test data to get the likelihood ratio of each pair. Prediction results obtained from the classifier were compared with actual labels to calculate the performance of model. In this implementation Joint Bayesian achieved 70% F1 score.

[6] https://github.com/rcmalli/keras-vggface

[7] https://www.microsoft.com/en-us/research/project/ms-celeb-1m-challenge-recognizing-one-million-celebrities-real-world/

The presented results have covered implementation, evaluation of one of the mentioned feature extractors with KNN/SVM and Joint Bayesian classifiers. And therefore solved the research question (Section 1, sub-section 1.2)

4.5 Implementation, Evaluation and Results of FaceNet Face Recognition Model

This section includes implementation of FaceNet model for face identification and face verification tasks, it covers objective 5.3. FaceNet system was presented by Schroff et al. (2015), which directly learns mapping from input face images to Euclidean space. In the Euclidean space, distances between feature vectors directly relates to the similarity of face images. In this model deep learning networks are trained to directly optimize the embeddings instead of intermediate layers. FaceNet was trained using triplets of anchor image, along with its matching and non-matching face patches. This approach achieved high representational efficiency and impressive face recognition performance with the help of only 128-bytes per face.

4.5.1 Implementation of FaceNet Feature Extractor

In this implementation pre-trained Keras FaceNet model was used. It was trained on MS-Celeb-1M dataset and 'facenet_keras.h5' was loaded in the keras model to get the pre-trained model for this implementation. Input layer of this network has shape (160, 160, 3) and output layer gives embedding as a 128-element vector. Face image dataset was pre-processed and reshaped as per the requirement of network. Feature vectors obtained from this network were later divided into training set and testing set for face identification and face verification tasks.

Face identification task was completed by implementing KNN and SVM classifiers and utilizing features extracted using FaceNet (section 4.5.2). For face verification task, Joint Bayesian classifier was implemented (section 4.5.3).

4.5.2 Implementation, Evaluation and Results of SVM and KNN Classifier on Features Extracted by FaceNet

SVM and KNN classifiers were used for face identification task. Feature vectors extracted using FaceNet network were divided into 80% for training set and 20% for testing set. Training set was used to train the classifiers.

KNN achieved 80% accuracy on the test dataset, whereas SVM achieved 80% accuracy. Identity cluster diagram for this model implementation is included in the configuration manual. It was observed that number of small identity clusters formed by the feature vectors extracted from this model was less than that of previous model (OpenFace), resulting low KNN accuracy than OpenFace model.

4.5.3 Implementation, Evaluation and Results of Joint Bayesian Classifier on Features Extracted by FaceNet

Face verification task was performed with the same approach implemented with above networks. Joint Bayesian network was used for the classification, to predict if input image pair depicts the same identity or not. Joint Bayesian model was trained with training set derived as 70% of embedding vectors. For testing, remaining embedding vectors portion was used and fed to trained Joint Bayesian network along with pair list. Pair list contained list of 6,000 positive pairs (i.e. face images of same identity) and 6,000 negative pairs (i.e. face images of different identities). Likelihood ratio obtained for each pair was compared with threshold to predict the similarity of input image pairs.

Prediction results were compared with actual labels to calculate the performance of model. In this implementation Joint Bayesian achieved 87% F1 score.

4.6 Implementation, Evaluation and Results of ResNet-50 Model fine-tuned with Equirectangular version data

This section includes details about fine-tuning process of pre-trained ResNet-50 model and covers objective 5.4. ResNet-50 model is based on deep residual learning technique. This technique adds few advantages for usage of ResNet in transfer learning: ResNet is easier to train; It is more tolerant of hyperparameters, including initial learning rate and regularization; It generalizes better. On the other hand, FaceNet and OpenFace models used in this project are based on triplet loss function. In these networks generating triplets for training model is a very crucial task, any wrong approach used for triplet generation can severely affects the training process. Fine tuning ResNet-50 with generated huge dataset was not possible in a single attempt, (due to requirement of high computational power, time and storage) hence fine-tuning was done by taking small batches of training dataset as discussed in the methodology section.

4.6.1 Implementation of ResNet-50 Model fine-tuned with Equirectangular version data

Implementation of this model basically includes 3 main steps: Pre-process generated equirectangular version of VGGFace2 data and upload in batches; Import and modify the ResNet-50 model as per the requirement; Set appropriate hyperparameters and train model. As per the first step pre-processing of the equirectangular version of VGGFace2 dataset was done with the steps mentioned in the methodology section. Before uploading the dataset batches on Google storage (for fine-tuning model on Google Colab) it was necessary to pre-process it locally in an optimized way. Dataset batch uploaded on Google storage was split into 80% training and 20% validation sets. ImageDataGenerator library by keras was used to create train_generator and test_generator with batch size 64 and class_mode categorical.

For the second step, ResNet-50 model was imported from keras_vggface library (library VGGFace), at the same time weights were loaded. This model was pre-trained on VGGFace2 dataset. In this architecture final layer was a classifier layer giving 8631-dimensional vector, In order to fine-tune the model this layer was removed and few dense layers were added: dense_1 (1024), dense_2 (1024), dense_3 (512), dense_4 (number of classes). First 3 layers were using relu activation while last one had softmax activation set. For each time this model was fine-tuned on new data batch, weights (from the previous training) were loaded before moving to the next step. For the very first batch of fine-tuning, all the layers of the model except last 4 newly created layers were set to freeze. For all subsequent batches only the last dense_4 layer was removed and replaced with a new dense layer (with dimensions same as number of classes utilized in respective data batch) and all except this newly created layer were set to freeze in later scenarios of fine-tuning.

Third step of this implementation was setting suitable hyperparameters, adding checkpoints and early stop functionality. Setting appropriate hyperparameters (e.g. learning rate, momentum) is extremely important for good performance. Since this implementation includes fine-tuning and not training the network from scratch learning rate was set to 1e-4.

Momentum was set to 0.9. As training dataset have more than two classes, loss was set to categorical_crossentropy. In the training process it was observed that improvement in validation loss and accuracy stopped after initial few epochs. To introduce early stop on achieving high accuracy, callbacks were introduced. For early stop validation loss was monitored to save weights after achieving high accuracy than previous epoch. After completion of fine-tuning using all batches, saved weight was downloaded on local machine and utilized for feature extraction on MOT-360 dataset. After extracting the feature vectors using this model, all were fed to the classifier for classification.

After feature extraction all the feature vectors were utilized by KNN and SVM classifiers to perform face identification (section 4.6.2) and utilized by Joint Bayesian classifier for face verification task (section 4.5.3).

4.6.2 Implementation, Evaluation and Results of SVM and KNN Classifier on Features Extracted by Fine-tuned ResNet-50

For face identification task, features extracted using fine-tuned ResNet-50 model were passed to SVM and KNN classifier. These classifiers were trained on 80% training data and tested on 20% testing data derived from extracted features.

KNN achieved 89% accuracy on the test dataset, whereas SVM had 92% accuracy.

4.6.3 Implementation, Evaluation and Results of Joint Bayesian Classifier on Features Extracted by Fine-tuned ResNet-50

Features extracted using ResNet-50 fine-tuned model were utilized by Joint Bayesian classifier, for face verification task performance evaluation. Face verification task was performed in the similar way as it was in previous implementations. Training of the classifier used 70% of the data and testing was performed on remaining 30% testing dataset along with programmatically created 6,000 positive pairs and 6,000 negative pairs. Likelihood ratios obtained as predictions were compared with the actual labels of testing dataset. For this feature extractor Joint Bayesian achieved 87% F1 score with 87% precision and 87% recall.

4.7 Comparison of Developed Models

This section includes comparison of performances of implemented models and covers objective 7. For face identification task, implemented feature extractors were compared with accuracy as evaluation metric. Here KNN and SVM both classifiers were used in order to consider distance-based simple technique as well as complex large margin classification method. Feature vectors extracted using OpenFace model got good accuracy with KNN as well as SVM classifier, followed by feature vectors from FaceNet model. However, ResNet-50 model has shown relatively poor performance. After fine-tuning this model with generated equirectangular version of VGGFace2 dataset accuracy in both KNN and SVM increased to 89% and 92% respectively; and outperformed all other models implemented in this project.

Table 2: Face identification performance results

CNN model	Accuracy (KNN)	Accuracy (SVM)
OpenFace	89%	85%
ResNet-50	79%	63%
FaceNet	84%	85%
ResNet-50 fine tuned	89%	92%

For face verification task, features extracted using CNN models paired up to form dataset containing positive pairs as well as negative pairs. As there were total 52 identities and Joint Bayesian classifier was trained on skewed classes (much more negative pairs than positive pairs). Therefore, F1 score was chosen as evaluation metric instead of accuracy. Comparison of Recall, Precision and F1 score results of implemented feature extractors are given in Table 3. These results were derived from respective results (Precision, Recall, F1 score) of negative pairs and positive pairs for each of the feature extractor. ResNet-50 has shown relatively poor performance in face verification task as compared to others. On the other hand OpenFace and FaceNet both have given equally good results, but slight difference in Precision. False positive rate in FaceNet was higher than that of OpenFace. ResNet-50 model after fine-tuning had remarkable improvement in F1 score, Precision, and Recall.

Table 3: Face Verification performance results

CNN model	Precision	Recall	F1 score
OpenFace	87%	84%	84%
ResNet-50	77%	71%	70%
FaceNet	87%	87%	87%
ResNet-50 fine tuned	87%	87%	87%

4.8 Conclusion

Results obtained from these implementations enable us to answer the research question. Models implemented in this section have all shown state-of-the-art results on different face image datasets. Since they were all pre-trained with huge face datasets, it was expected from them to deliver good results on MOT-360 dataset too. These models met the expectations with reasonably good results. In the case of fine-tuning of ResNet-50 model, improvement in its results were also observed.

5 Discussion

As discussed in the methodology section, Analysis of spherical images comes with various challenges and dealing with distortions is a major challenge in this process. Distortion and discontinuity issues are the results of transformation of the spherical image to its planar representation. Although there are few new representations such as SpherePHD which solves the distortion issue up to some extent, many researches have chosen ERP representation techniques due to its simplicity and being less costly to implement. Representations like SpherePHD plays important role when the problem needs complete image for analysis and, problems such as object detection (Lee et al. 2019). Similarly, in the case of face recognition, these representations will be crucial for face detection as less distorted face images will be present. But only after successfully detecting a face from spherical image further analysis on face image can be done which includes face identification and face verification. By this logic ERP representation was chosen for this project as the scope of this project was limited to face identification and face verification only (excluding face detection).

Dataset used in this project had ample of face images with severe distortions and very few of these images were detected by used mtcnn predictor. The criteria used by mtcnn or other strict face predictors for face detection includes landmarks such as outer corners of both the eyes and mouth, which was main reason of failure in face prediction in distorted images. To handle this issue a relaxed face predictor was used in this project, which considered only outer corners of both eyes and nose as landmark. Even after using this relaxed version of face predictor only a few faces were predicted. This could be the limitation of this project.

6 Conclusion and Future Work

This project tried to answer, "*Can feature extractor models (FaceNet, OpenFace, ResNet-50, and ResNet-50 fine-tuned with spherical face images) for face images enhance/improve face identification and verification in spherical images?*" by implementing various face recognition models and perform face recognition tasks on spherical images dataset. Project objective also included generation of equirectangular version of VGGFace2 dataset and subsequently use it for fine-tuning one of the models. In this project OpenFace, FaceNet and ResNet-50 models were implemented for feature extraction. Along with these pre-trained networks, a ResNet-50 network fine-tuned with generated equirectangular version of VGGFace2 dataset was also implemented for feature extraction. Performances of implemented models were compared to find the best model among the implemented models for face recognition in spherical images task. For better performance various pre-processing steps were included in the project and performed on both MOT-360 and equirectangular version of VGGFace2 dataset. As per the results, it was observed that models FaceNet and its variant OpenFace performed better as compared to ResNet-50. In overall all the implemented models delivered reasonably fine results on both face identification (classification using KNN and SVM) and face verification (binary classification using Joint Bayesian) tasks. Moreover, it was observed that fine-tuning the feature extractor (ResNet-50) improved its performance.

As mentioned in the literature review there has been lot of on-going research related to analysis of spherical image. Networks like Spherical CNN, Graph-based CNN are specifically designed for analysis of spherical images, majorly focusing on tasks like object-detection and image classification. Currently implementation of these networks for face recognition was out of the scope of this project, but for future work these networks can be implemented and fin-tuned with spherical face image dataset to evaluate their performance in face recognition domain. Due to time constraint and restrictions as well, not all the images on Google Colab platform from equirectangular version of VGGFace2 dataset were utilized for fine-tuning ResNet-50 model. For future work more data can be utilized for fine-tuning the model.

7 Acknowledgement

Many Thanks to my supervisor Dr. Catherine Mulwa for the guidance and support throughout the semester. I would also like to thank Dr. Marcos Cirne at the Institute of computing, UNICAMP, Brazil for the guidance and help.

References

Amos, B., Ludwiczuk, B. & Satyanarayanan, M., 2016. *Openface: A general-purpose face recognition library with mobile applications.*

Cao, X. et al., 2013. *A Practical Transfer Learning Algorithm for Face Verification.* Sydney, NSW, IEEE International Conference on Computer Vision, pp. 3208-3215.

Chen, J., Patel, V. M. & Chellappa, R., 2016. *Unconstrained face verification using deep cnn features.*, IEEE Winter Conference on Applications of Computer Vision, pp. 1-9.

Cirne, M. et al., 2019. *Deep Face Verification for Spherical Images.* Taipei, Taiwan, IEEE International Conference on Image Processing (ICIP), pp. 3292-3296.

Cohen, T. S., Geiger, M., Koehler, J. & Welling, M., 2018. *Spherical CNNs.,* International Conference on Learning Representations (ICLR).

Coors, B., Condurache, A. P. & Geiger, A., 2018. *Spherenet: Learning spherical representations for detection and classification in omnidirectional images*., The European Conference on Computer Vision (ECCV).

Frossard, P. & Khasanova, R., 2017. *Graph-Based Classification of Omnidirectional Images.,* International Conference on Computer Vision Workshops (ICCVW), pp. 860-869.

He, K., Zhang, X., Ren, S. & Sun, J., 2016. *Deep Residual Learning for Image Recognition.* Las Vegas, NV, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.

Hu, G. et al., 2015. *When Face Recognition Meets with Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition*., ICCV.

Lai, W. et al., 2018. *Semantic-Driven Generation of Hyperlapse from 360 Degree Video*., IEEE Transactions on Visualization and Computer Graphics, pp. 2610-2621.

Lantz, B., 2013. *Machine learning with R.* s.l.:Packt Publishing Ltd.

Lee, Y. et al., 2019. *SpherePHD: Applying CNNs on a Spherical PolyHeDron Representation of 360° Images.* Long Beach, CA, USA, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9173-9181.

Lin, L. et al., 2018. *Active Self-Paced Learning for Cost-Effective and Progressive Face Identification.* s.l., IEEE Transactions on Pattern Analysis and Machine Intelligence.

Lu, C. & Tang, X., 2014. *Learning the face prior for bayesian face recognition*., European Conference on Computer Vision, pp. 119-134.

McLaughlin, M. R. & Herlocker, J. L., 2004. *A collaborative filtering algorithm and evaluation metric that accurately model the user experience.* s.l., Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, p. 329–336.

Monroy, R., Lutz, S., Chalasani, T. & Smolic, A., 2018. *SalNet360: Saliency Maps for omni-directional images with CNN.*

Paisitkriangkrai, S., Shen, C. & Hengel, A. v. d., 2015. *Learning to rank in person re-identification with metric ensembles*., CVPR.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016. *You Only Look Once: Unified, Real-Time Object Detection*.

Schroff, F., Kalenichenko, D. & Philbin, J., 2015. *Facenet: A unified embedding for face recognition and clustering.*, IEEE Conference on Computer Vision and Pattern Recognition, p. 815–823.

Simonyan, K., Parkhi, O., Vedaldi, A. & Zisserman, A., 2013. *Fisher Vector Faces in the Wild*., British Machine Vision Conference.

Sun, Y., Wang, X. & Tang, X., 2014. *Deep Learning Face Representation from Predicting 10,000 Classes.* Columbus, OH, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1891-1898.

Tateno, K., Navab, N. & Tombari, a. F., 2018. *Distortion-aware convolutional filters for dense prediction in panoramic images*., the European Conference on Computer Vision (ECCV), p. 707–722.

Wan, L., Liu, N., Huo, H. & Fang, T., 2017. *Face Recognition with Convolutional Neural Networks and subspace learning.* Chengdu, 2nd International Conference on Image, Vision and Computing (ICIVC), pp. 228-233.

Yu-Chuan Su, K. G., 2018. *Learning Spherical Convolution for Fast Features from 360° Imagery*.

Zhao, Q. et al., 2018. *Distortion-aware CNNs for Spherical Images.*, pp. 1198-1204.