# B.TECH. PROJECT REPORT

## On

# An Efficient Approach to Match Highly Similar 3D Objects

## BY

**Arushi Jain, 160001008**
**and**
**Rotte Priyanka Ajay, 160001051**


**DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY INDORE**

**December, 2019**

**PROJECT REPORT**

*Submitted in partial fulfillment of the*
*requirements for the award of the degree*

***of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

*Submitted by:*

**Arushi Jain, 160001008**
**and**
**Rotte Priyanka Ajay, 160001051,**

**Discipline of Computer Science and Engineering,**

**Indian Institute of Technology, Indore**

*Guided by:*

**Dr. Surya Prakash,**

**Associate Professor,**

**Computer Science and Engineering,**

**IIT Indore**


INDIAN INSTITUTE OF TECHNOLOGY INDORE
December, 2019

## Candidates' Declaration

We hereby declare that the project entitled "An Efficient Approach to Match Highly Similar 3D Objects" submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering completed under the supervision of Dr. Surya Prakash, Associate Professor, Computer Science and Engineering, IIT Indore is an authentic work.

Further, we declare that we have not submitted his work for the award of any other degree elsewhere.

**Arushi Jain**
(160001008)
**05/12/2019**

**Rotte Priyanka Ajay**
(160001051)
**05/12/2019**

---

## Certificate by BTP Guide

It is certified that the above statement made by the students is correct to the best of my knowledge.

**Dr. Surya Prakash**
Associate Professor
Computer Science and Enigneering
IIT Indore

# Preface

This report on "An Efficient Approach to Match Highly Similar 3D Objects" is prepared under the guidance of Dr. Surya Prakash.
This report aims at explaining our work in the field of Computer Vision and Image Processing to develop a novel deep learning model that can classify similar 3D objects such as 3D faces. We present our motivation and approach towards the problem, the concepts used in developing the algorithm and its implementation details. We have tried to the best of our abilities and knowledge to explain the content in a lucid manner.

**Arushi Jain**
B.Tech IV year
CSE
IIT Indore

**Rotte Priyanka Ajay**
B.Tech IV year
CSE
IIT Indore

# Acknowledgements

We wish to thank Dr. Surya Prakash for his kind support and valuable guidance. He motivated us constantly to work harder and explore more in our project area. It was through his dedicated and adept insight which saw us through various roadblocks and difficulties. We are also grateful to Mr. Akhilesh Mohan Srivastava for his help throughout the BTP.

**Arushi Jain**
B.Tech IV year
CSE
IIT Indore

**Rotte Priyanka Ajay**
B.Tech IV year
CSE
IIT Indore

# Abstract

In this work, we develop a generic tool for the recognition of similar objects. The techniques proposed for object recognition mainly focus on categorizing heterogeneous objects. However, when subjected to the multi-class classification problem of similar objects, these models don't fare so well. PointNet architecture is one such model that directly consumes point clouds of images as inputs, instead of relying on the much bulkier 3D voxels and grids, to classify multiple objects. Even though it gives remarkable accuracy when classifying different objects, the performance declines when we try to classify similar objects like faces. We are proposing a solution that combines the object classification utility from PointNet architecture along with One-Shot Learning from Siamese Network that converts our multi-class classification problem to a binary classification problem and improves object recognition accuracy, even for similar objects. We are applying our proposed approach on 3D face recognition by conducting a series of experiments on three 3D face databases, namely, IIT Indore database, Bosphorus database, and University of Notre Dame (UND) database, to test our model. We also use a novel data augmentation technique that uses sub-sampling from the existing point clouds to increase the size and variability of the available data. The experimental results show that the proposed method is considerably better in recognizing objects that are highly similar as compared to the original PointNet architecture.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

One of the most active research fields in Computer Vision is 3D object recognition. Recognition is the first step in the semantic analysis of an object. The main objective of object classification is to recognize previously unknown objects in digital images and 3D spaces. Object recognition techniques typically use matching, learning, or pattern recognition algorithms using techniques based on appearance or features. With the advent of new algorithms, models, and approaches, 3D object recognition is becoming increasingly effective. The manufacturing industry, autonomous driving, video surveillance, urban planning, control and safety, and augmented reality extensively use 3D object classification and recognition. Most of the existing 2D and 3D use convolutional neural networks (CNNs), which successfully extract features from the data.

Even though most of the proposed object recognition techniques successfully classify and recognize unknown objects, they are not very successful when we apply them on subjects belonging to the same object class. We are proposing a tool that is a unique combination of one of the popular object recognition architectures and a one-shot learning network to overcome the above-mentioned shortcoming. Face recognition can be one of the use cases of object recognition for objects belonging to the same class. It is a widely adopted biometric technology for access control in security, primarily due to its contactless and non-invasive nature, unlike fingerprints and iris recognition. Due to its limitations in adapting to changes in parameters like illumination and poses such as emotions and occlusions, the research focus is gradually being shifted from 2D face recognition to 3D face recognition.

There are two utilities of face recognition, namely, face verification, which is a 1:1 comparison, and face identification, which is a 1:N juxtaposition problem. In recent years, major developments in face recognition techniques have increased the face recognition performance by manifold. However, most

of the techniques face the common challenge of the variations in the face acquisition process, such as illumination irregularities due to reflection and pose changes due to varied expressions, deformations, or occlusions. These differences are more pronounced in 2D face recognition as compared to 3D face recognition. This is primarily because 3D based approaches use the entire face geometry data. There are two common approaches used to model 3D face data - 2.5D depth images that represent 3D points in a 2D space along with different viewpoints, and 3D images that globally represent the entire face geometry, for example, 3D voxels and point clouds.

Most of the deep learning-based approaches on 3D data use volumetric CNNs in which the input is highly regularized in the form of 3D voxels or image grids. This representation simplifies weight sharing and other kernel optimizations. However, 3D voxels and grids are bulky in nature, thus rendering the input computationally and spatially expensive. To overcome this drawback, our approach directly consumes the input as an unordered set of data points called point cloud instead of transforming them into a regular 3D representation. 3D point cloud data processing is an important research field in computer vision with applications in object classification and recognition, environmental detection, mobile robot navigation, and so on. Point clouds have an added advantage of invariance to transformations like translation and rotation.

As discussed in [1], PointNet architecture is a deep learning CNN framework that directly consumes point clouds as input. It is an efficient model in classifying 3D dissimilar objects and is robust with respect to input disturbances. The uniqueness of the PointNet architecture lies in its ability to preserve the translational and rotational invariance property of point clouds. However, PointNet architecture is used to classify inputs into different object classes rather than different samples of the same class. Thus, face recognition problems cannot be solely solved by deploying this architecture alone.

Thus to improvise over the PointNet architecture model, we're combining it with the Siamese network. The Siamese Network generates a similarity score by matching a test sample with a reference sample and predicting whether they belong to the same subject class or different subject classes. This similarity score lies in the range 0, indicating no similarity, to 1 indicating full similarity. The similarity function of the Siamese network thus takes in two inputs and expresses how similar they are to each other.

One of the limitations of 3D object recognition is the difficulty in acquiring a large amount of 3D data. Due to this limitation, effectively training the model without overfitting is a challenge. In our work, we are using a novel augmentation technique to increase the size of our database before training

the model. We are creating different fixed-size subsets of the input point clouds of our samples. These subsets will have fewer points than the original sample, reducing the computational requirements but still maintaining all the features of the original sample.

The rest of the report is organized as follows: Chapter 2 presents literature review of 3D Object Classification and 3D Face Recognition Techniques. Chapter 3 discusses preliminaries required for the proposed technique. Our proposed model is described in Chapter 4. The results of the analysis and experiments on the proposed method are presented in Chapter 5. Chapter 6 concludes the report.

# Chapter 2

# Literature Review

Previously, 3D object identification used approaches like Iterative Closest Point [2], the differential geometry approach [3] and calculating free-form curved surfaces using spherical correlation [4]. Deep learning-based approaches are a class of machine learning algorithms that extracts high-level features from raw input using multiple layers of representation and abstraction. Convolutional Neural Network is a type of deep learning model that extracts complex features from raw images using local pooling and filters. They have been particularly useful in character recognition [5] and EEG signal recognition [6]. [7] proposes a multi-scale 3D deep convolutional neural network for hyperspectral image Recognition. [8] proposes a slice-based CNN approach to recognize 3D objects in real-time, achieving a success rate of 94.34% in ModelNet10 Recognition. To address the problems of inter-class similarities, intra-class variances, and spatial variability in images, [9] proposes a framework for object recognition that is discriminative and spatially invariant.

Robotic perception and manipulation also requires applications of 3D object detection and pose estimation. This is done by generating a synthetic dataset from its 2D and 3D local features [10]. Sales et al. [11] propose a novel 3D shape descriptor for recognizing objects in 3D scenes which is taken as input for a supervised machine learning model. In order to fully utilize the volumetric information which is usually hidden in the depth data, a view-based 3D model is constructed from a single depth image [12]. 3D feature information is often lost while being converted to their respective voxel representations. To overcome this, a new rotation-invariant feature [13] is proposed which is based on mean curvature. This method improves the recognition rate on voxel CNNs and increases the overall accuracy on ModelNet 10 dataset by 1%. Another 3D object detection approach involves a system with an enhanced Depth Estimation Algorithm [14] which makes use of statistical calculations for refining the depth image and reduces the

effect of noise.

Biometrics is one of the major applications of 3D object recognition. Due to their unique anatomical markers, 3D ear and face structures can be used to create strong biometric security systems. One of the earlier approaches involved Principal Component Analysis to extract features from 3D ear surface for 2D ear image identification [15]. Ganapathi et al. [16] propose a technique that uses 2D and 3D ear images for biometric recognition using local feature detection and description. The proposed approach achieves a remarkable accuracy of 98.69% on UND J2 dataset.

Yi Sun et al. [17] achieve an accuracy of 96% on the 2D LFW dataset by combining two CNNs derived from convolution and inception layers. Another deep CNN approach, proposed by Jun-Cheng Chen et al. [18], trains real-world unconstrained 2D LFW faces and achieves an accuracy of 97.45%. For 2.5D depth images, Lv et al. [19] get a recognition rate of 97.8% by using LBP for feature extraction and sparse representation classifier on FRGCv2.0. Extracting accurate facial landmarks is one of the obstacles in processing 3D faces. [20] studies facial shapes using scanned 3D images and analyzes different approaches to extract facial landmarks. [21] gives a detailed overview of recently used 3D face recognition algorithms, databases, features, and associated challenges due to variations in expressions, poses, and occlusions. [22] presents an efficient 3D face recognition approach to address the problem of partial data like corrupted data, occlusions, and single training sample.

Most of the 3D face recognition algorithms that use point clouds try to solve expression variations, but very few have been successful in solving challenges caused by pose changes and occlusions. [23] extends the SIFT-like matching framework to mesh data and proposes an approach that uses fine-grained matching of 3D keypoint descriptors. [24] presents a comprehensive parametric study of two CNN models on face recognition. The respective models differ in combinations of activation functions, learning rates, and filter size. The saturation due to the limited gallery size of 3D databases has hindered the development in the field of 3D biometrics, despite recent developments in deep learning. [25] proposes a method for generating a large corpus of labeled 3D face scans for training and a solution to merge existing 3D databases for testing. [26] proposes a Deep CNN and a 3D augmentation technique that synthesizes a number of different facial expressions from a single 3D face scan.

Most of the Deep Learning-based approaches to 3D data use Volumetric CNNs. [27, 28, 29] uses voxelized shapes as inputs to 3D CNNs. However, such representation is hindered by sparse data spaces and computationally expensive convolution operations. Capturing fine facial structures requires

a very high voxel resolution, thus consuming massive amounts of memory. Point Cloud Features encode the given set of 3D points such that they are invariant to certain intrinsic [30, 31] and extrinsic [32] transformations. These features can be local or global, which need to be combined optimally to get the best possible models. 3D data in the form of vectors are used by Feature-based DNNs [33, 34] that extract original features of the shapes and classify those shapes using a fully connected network.

Deep Siamese Neural Networks have also been used for face recognition [35]. [36] presents one such approach based equipped with supervised loss function, which increases the inter-class variations by maximizing the distance between the features for different classes while minimizing the intra-class variations. [37] also uses Siamese Network to match scanned facial images with digital ones.

# Chapter 3

# Preliminaries

In our proposed method, we are combining the PointNet Architecture with the Siamese Network. PointNet is a unified architecture used for various applications such as object classification and part segmentation. It is a highly efficient and robust architecture that gives strong performance for dissimilar objects. We're extending PointNet architecture to effectively match similar 3D objects by adding the Siamese network on top of it.

## 3.1 PointNet Architecture

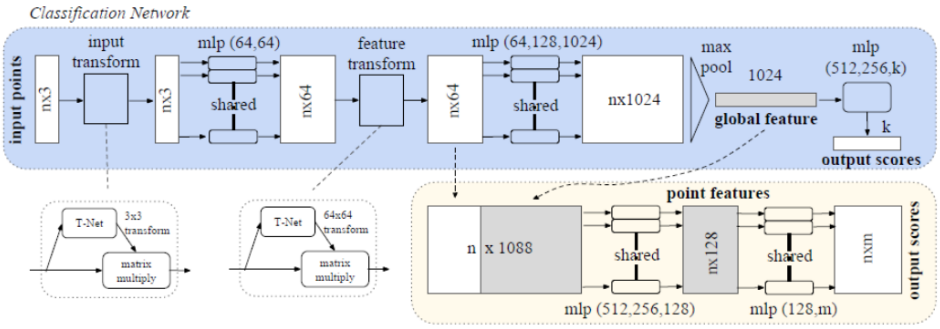



Figure 3.1: PointNet Architecture (Figure taken from [1])

The PointNet architecture [1] is inspired by three essential properties of the point clouds. First, being a set of points, the point clouds are invariant to their N! Permutations. Hence, they are unordered. Second, there are interactions among points, meaning that in spite of being in a set, the neighboring points in the space form meaningful subsets that represent a local structure. Lastly, the point clouds are invariant under transformations. Rotating or translating the points of a point cloud altogether does not mod-

ify the point cloud itself as it is a geometric object.

The PointNet architecture mainly consists of three modules, a max-pooling layer, a local and global information combination structure, and two joint alignment networks. The max-pooling layer is used to aggregate the information from all the points. The local and global information combination structure is used in part segmentation. The structure first computes the global feature vector and feeds the global feature vector back to the point features by joining it with each of these point features. Now, when the new point features are extracted, they have the local as well as the global information in them. The first joint alignment network aligns the input points while the second network is used to align the point features generated by the architecture. An affine tranformation matrix is predicted by the network which is directly applied to the input point clouds, thus aligning all input sets to a canonical space before extracting features. The same alignment technique is extended for aligning the feature space. The two joint alignment networks are used to maintain the geometric invariance property of the point clouds. The complete architecture is shown Figure 3.1.

The PointNet architecture provides a effective way for multi-class object classification. But, for classifying objects belonging to the same object class, we require a more robust solution on the top of this architecture.

## 3.2 Siamese Network

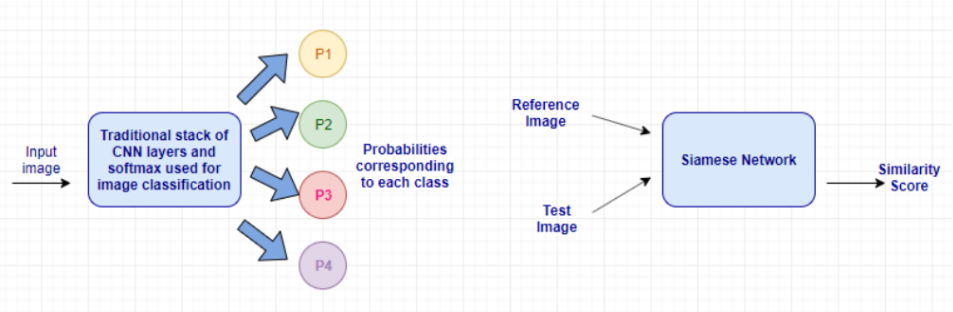



Figure 3.2: Traditional CNN Vs Siamese Network

In standard classification problems, a probability distribution over all the classes is generated after feeding the input image to a series of layers. However, the Siamese network uses a similarity score as visualized in Figure

3.2, between the test image and a reference image to check if they belong to the same or different class based on a threshold value to train itself. Here, the reference image is a precomputed vector forming a baseline against which the test vector is compared.

Instead of directly classifying a test image into one of the available classes, the Siamese network, by taking references from each class, produces a similarity score that represents the possibility that the test and the reference images belong to the same class. This similarity score lies in the range 0 to 1 where 0 indicates no similarity and 1 indicates full similarity. Thus, the Siamese Network learns a similarity function which takes in two inputs and expresses how similar they are to each other. Hence, Siamese network is a one-shot learning technique. This avoids using a large number of samples for training as well as eliminates the need to retrain the model when new classes are introduced.

# Chapter 4

# Proposed Methodology

In our proposed method, we are representing the 3D objects as a set of 3D points called point clouds. We use a novel augmentation technique to augment our 3D data, which will increase the size of our otherwise limited database, making the training more robust. The point cloud of each object is used as input to the PointNet architecture. After the model is trained, we extract features from the second-last dense layer of the architecture. The extracted features are then used to train the Siamese network, which finally predicts whether two objects belong to the same class or different classes. Our method is a generic approach for 3D object classification which we are applying for 3D face recognition. Figure 4.1 shows our proposed object recognition process.

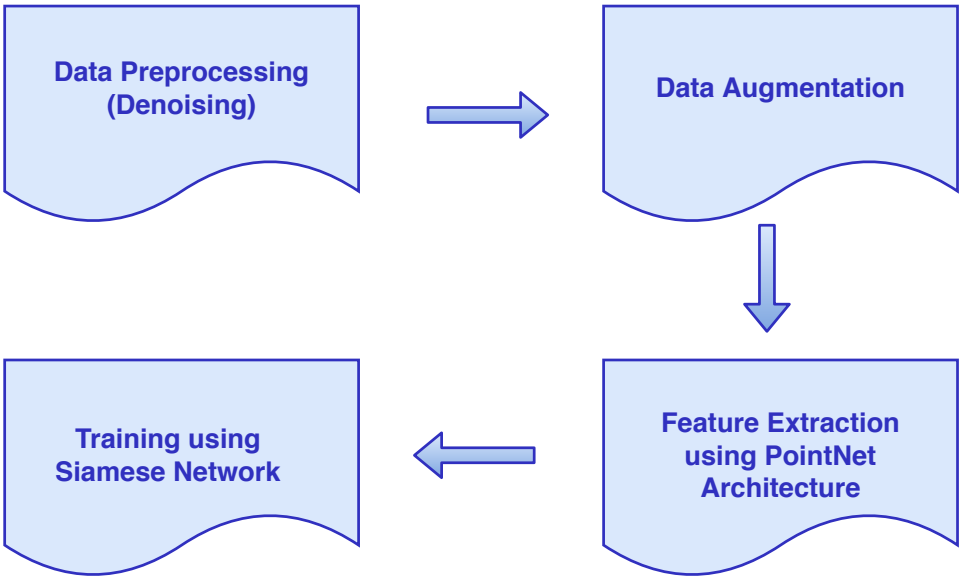



Figure 4.1: Proposed object recognition process

## 4.1 Preprocessing

Preprocessing of 3D scans is required to eliminate variations in poses as well as 3D noise, which can impact the performance of our object classification model. Generally, this elimination is achieved using frontalization and denoising techniques, respectively.

The 3D objects to be classified need to be aligned to a base reference image to make our database uniform in accordance with some ground truth. Algorithms such as ICP [2] can be used to minimize the distance between the objects in the database and the reference object. However, as discussed in Section 3.1, the PointNet architecture is invariant to geometric transformations of rotation and translation; thus, eliminating the need to frontalize the objects.

The point clouds of the 3D objects can get contaminated with spikes due to sensor noise, thus affecting the feature extraction process. We use a spike removal technique, where a sliding window is moved across the object, and the offset along the Z-coordinates is calculated, which, if greater than a threshold value, translates the center of the window to the mean offset. This process denoises the 3D scan, where the spikes are limited by the given threshold.

## 4.2 Augmentation

There is very limited data available for 3D objects. Due to only a few samples at hand for each subject, it is not possible to effectively train the model. The model learns the details and noise of these few samples so well that it negatively impacts the testing of this model on new data. To avoid this problem of overfitting, we increase the variability of our data by enlarging the database using a novel augmentation technique.

Our 3D input data is in the form of point clouds that contain an unordered set of 3D points, as explained in Section 3.1. However, we cannot use the entire point cloud of a single sample as input because of the computational and memory limitations involved. As a solution, we create different unordered subsets from the points in the point cloud of the sample. Each subset contains a fixed number of points, which is approximately 50% of the total points in the point cloud. Figure 4.2 shows one such example of a 3D face scan. The original face contains around 55000 points in the point cloud. We are randomly creating seven different subsets of 25000 points each. We can see from the figure that the overall geometry and the structure of the

face is maintained even after reducing the number of points in the point cloud by approximately half. Our method also ensures that the information from the entire point cloud is getting utilized without overshooting the computational and memory requirements.

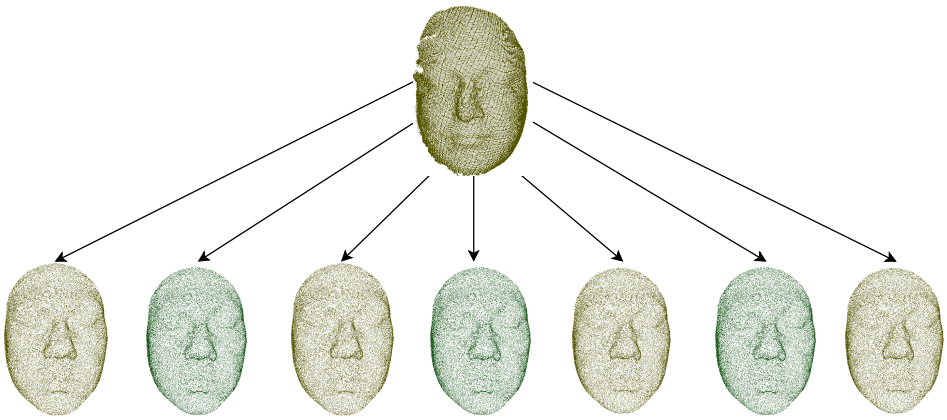


Figure 4.2: An example of the proposed augmentation technique

Further, we validate our above solution by calculating a similarity score as discussed later in Section **??** between the subsamples created from the same original sample. From the similarity scores shown in Figure 5.1, we observe that the subsamples show a stark similarity with each other, implying that the features remain intact even after scaling down the number of points. Using this novel technique, we create two different kinds of augmented databases, as explained below:

- `Random point cloud augmentation (Type I)`: In this method, we randomly select a specific number of 3D points from the entire point cloud of the 3D scans repeatedly to generate multiple samples from a single sample. We do this in a round-robin fashion for each subject to create a uniform number of augmented samples. However, individual samples may or may not be uniformly used.

- `Random point cloud augmentation (Type II)`: This technique differs from the previous one in the sense that every sample for each subject is used uniformly to create the augmented data. However, the number of total augmented samples for each subject may or may not be the same.

There exist various other augmentation techniques such as geometric transformations of translation and rotation and kernel filters. Transformations are not useful, since PointNet architecture is invariant to affine transformations as discussed in Section 3.1, thus rendering duplicate copies of the

3D scans. Also, as our model uses CNN architecture, kernel filtering is inherently used while training.

## 4.3 Proposed Model

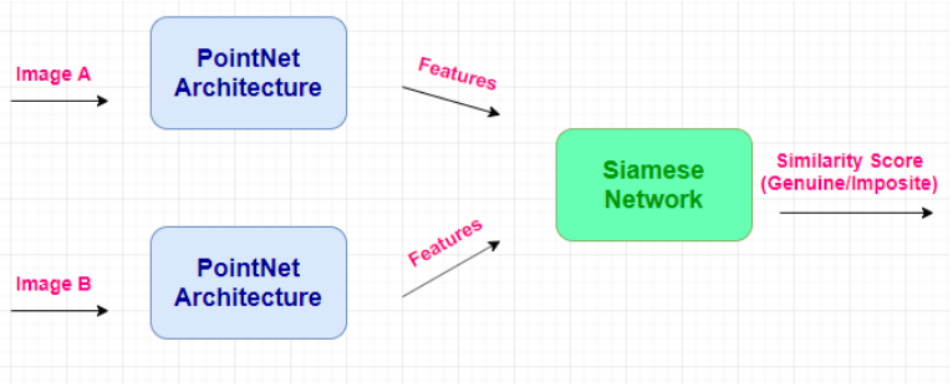



Figure 4.3: Overview of the proposed architecture

Our proposed model combines PointNet Architecture with Siamese Network. We use PointNet Architecture for feature extraction and Siamese Network for the classification based on these extracted features. The overview of our proposed architecture is shown in Figure 4.3.

First, we split our database into two sets - the train set and the test set. PointNet Architecture is trained over the train set of our data. Typically, PointNet gives the class of the input sample as the output of its last layer. Instead, we are using PointNet till its second-last dense layer. This layer outputs the feature vectors for the given samples. We extract these features of our train set to then train the Siamese Network.

For Siamese Network, we need two sets of pairs from our extracted features - genuine pairs that contain features from the samples belonging to the same class and imposite pairs, which contain features from the samples belonging to different classes. The Siamese Network gets trained on these feature pairs to predict whether it is genuine or imposite. We create all possible genuine pairs for each class and all possible imposite pairs by taking combinations of each class with every other class to make the training more robust.

Finally, we test our model by first passing our test set through the trained PointNet Architecture that generates feature vectors for the test set and then compares each of these test features with train features of all classes to

predict whether the combination is genuine or imposite. By pairing the test image with a reference from each class, the Siamese network calculates a similarity score for each pair and predicts the subject class with the highest similarity score as the class of the test image. The detailed architecture of our proposed model is shown in Figure 4.4.

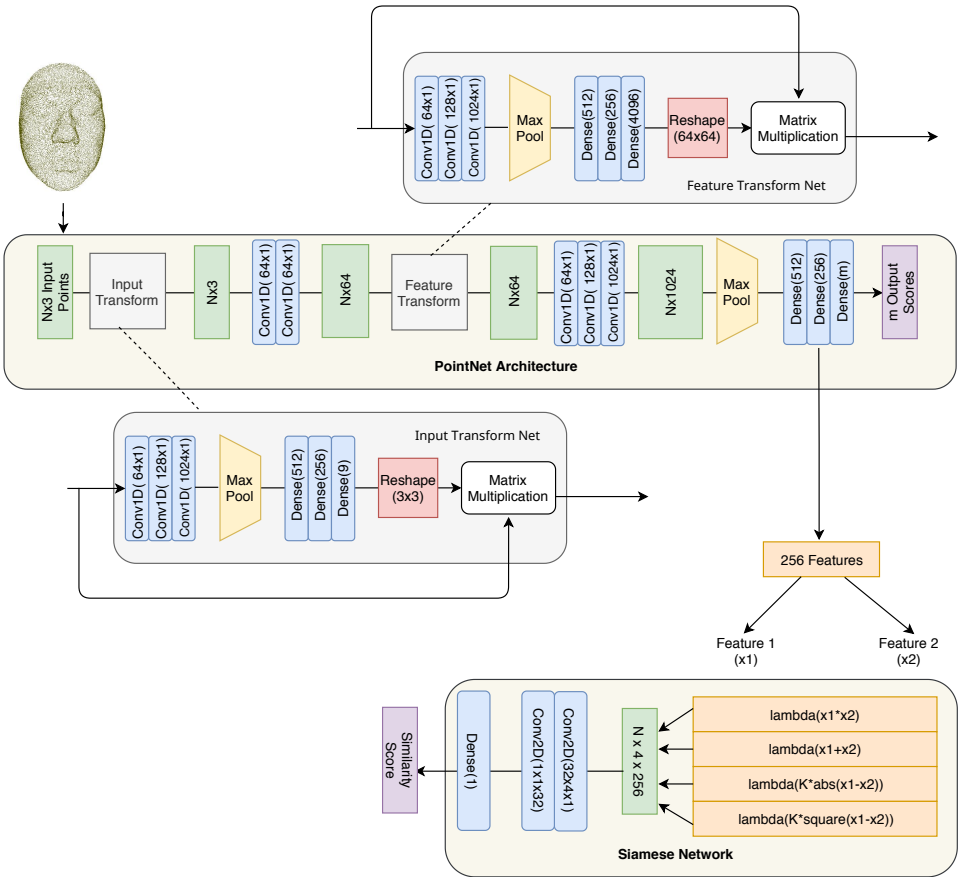



Figure 4.4: The Proposed Model

# Chapter 5

# Experiments and Results

We now apply our proposed object recognition model for 3D face recognition. We use three 3D face databases to test our model - IIT Indore (IITI) database, Bosphorus database, and University of Notre Dame (UND) database. We expand our database to improvise feature extraction and prevent overfitting by augmenting our databases. These databases are randomly split into two sets: 70% for training and 30% for testing. We evaluate classification performance based on recognition accuracy.

## 5.1 3D Databases Used

**IITI** The IIT Indore database contains 170 subjects with a total of 445 samples. An Artec 3D EVA scanner was used to acquire these 3D face scans. The samples in the database are unaligned. However, we do not need to align the images since PointNet architecture is invariant to geometric transformations.

**Bosphorus** The Bosphorus database [38] contains 105 subjects with 4666 3D facial scans. The database has a rich set of expressions, systematic pose variations and various occlusions. Out of these, we are testing our model on the 299 neutral faces to compare the performance with other databases. An Inspeck Mega Capturor II 3D was used to acquire this facial data. The face scans in the database are aligned and contain minimal noise as the noise reduction is already done at the time of data acquisition by experimentally optimizing the acquisition setup.

**UND** The University of Notre Dame database (ND-collection D) [39, 40] contains 277 subjects with 953 aligned 3D face scans. A Minolta Vivid 900 3D range scanner was used to acquire these images. These face scans contain considerable noise in the form of spikes. Hence, we are using spike removal technique to denoise the 3D scans.

## 5.2 Augmentation of 3D Databases

Since the number of samples per subject is quite low for the given databases, we augment them to increase the sample size. As discussed in Section 4.2, we are applying two types of random point cloud augmentation. Type I augmentation creates 21 samples per subject and Type II augmentation creates 21 samples for each original sample present in a subject. The total number of samples after each type of augmentation is shown in Table 5.1.

| Database | No. of subjects | Original no. of samples | No. of samples after Type I augmentation | No. of samples after Type II augmentation |
|---|---|---|---|---|
| IITI | 170 | 445 | 3570 | 9345 |
| Bosphorus | 105 | 299 | 2205 | 6279 |
| UND | 277 | 953 | 5817 | 20013 |

Table 5.1: 3D face databases

In the Siamese Model, similarity scores are calculated such that the imposite pairs have score closer to 0 and genuine pairs have score closer to 1. When we create multiple samples from the same sample using Random Point Cloud Augmentation, we need to validate that the features of the original samples are not compromised. From Figure 5.1, we can see that the Similarity Scores for $i^{th}$ and $(i+1)^{th}$ samples within the same subject class for 50 subjects are close to 1, thus implying that the similarity is maintained in spite of the reduction in points.

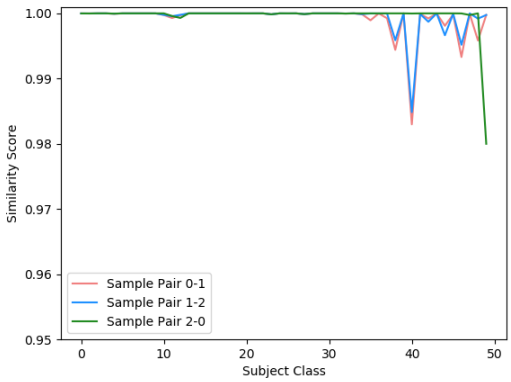



Figure 5.1: Similarity scores for sample pairs within the same subject class

## 5.3 Performance on 3D Databases

We split each of the IITI, Bosphorus and UND databases into a 70-30 ratio to create train sets and test sets. Table 5.2 gives the training accuracy on the train set when it is passed through the Pointnet architecture:

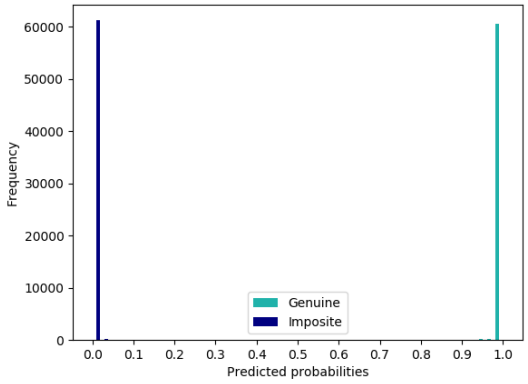
| Database | Accuracy for Type I Augmentation | Accuracy for Type II Augmentation |
|---|---|---|
| IITI | 87.5 | 88.86 |
| Bosphorus | 84.6 | 83.97 |
| UND | 79.95 | 77.08 |

Table 5.2: Results on the PointNet Architecture

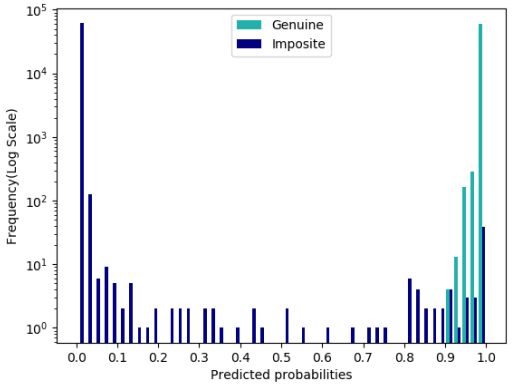
For the Siamese Model, the distribution of the similarity scores for imposite (Class 0) and genuine (Class 1) classes for each of the 3 databases is shown in Figure 5.2. Since maximum concentration of similarity scores occurs near 0 (for an imposite pair) and 1 (for a genuine pair), we also plot the distribution on log scale to show the number of samples in the intermediate range. It is evident from the graphs that the similarity scores are quite accurate for both genuine and imposite classes i.e. the score for most of the imposite pairs is close to or equal to 0 and that for genuine pairs is close to or equal to 1.

To find an accurate decision boundary for a pair to be classified as either imposite or genuine, we plot a threshold vs FAR-FRR curve for each of the 3 databases as shown in Figure 5.3 and arrive at the given values for the thresholds:- IITI - 0.97, Bosphorus - 0.88 and UND - 0.68.

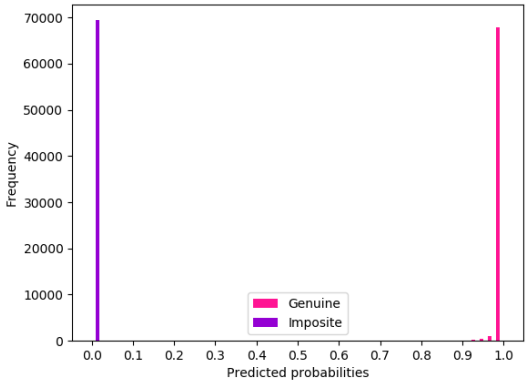
We use 70-30 train-test split for creating genuine and imposite pairs for the Siamese Network i.e. 70% of the total pairs are created from train features obtained from PointNet Architecture and remaining 30% are created from the test features extracted from the trained PointNet Architecture. After training the Siamese network on these 70% of the pairs, we can test our model in two ways. One way is to create pairs such that one sample is from the test set and the other is from the trained set (test-train pairs). This testing shows how well our model compares an unknown sample with the known samples to determine the unknown sample's subject class. Another way is to create pairs from the test set itself (test-test pairs) to examine our model's ability to recognize genuine and imposite pairs from the unknown set of samples.



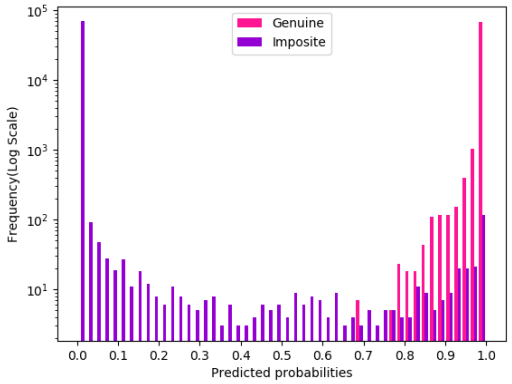

(a) Distribution for IITI database



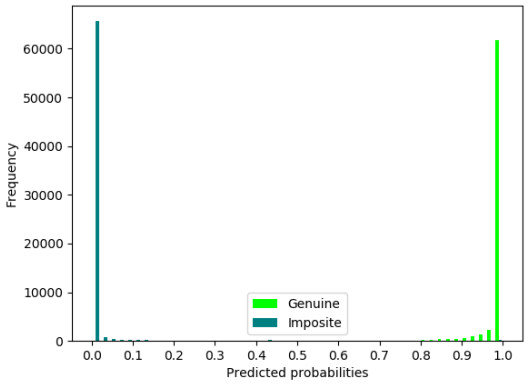

(b) Distribution (Log Scale) for IITI database



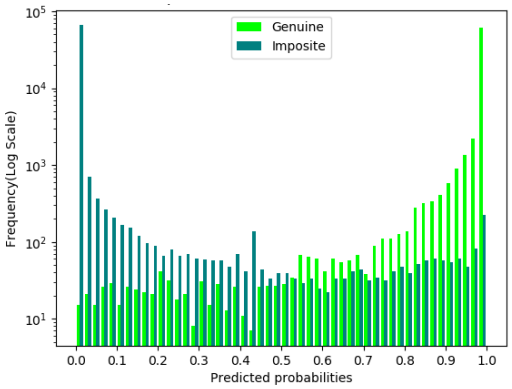

(c) Distribution for Bosphorus database




(d) Distribution (Log Scale) for Bosphorus database




(e) Distribution for UND database




(f) Distribution (Log Scale) for UND database

Figure 5.2: Similarity score distribution for IITI, Bosphorus and UND databases

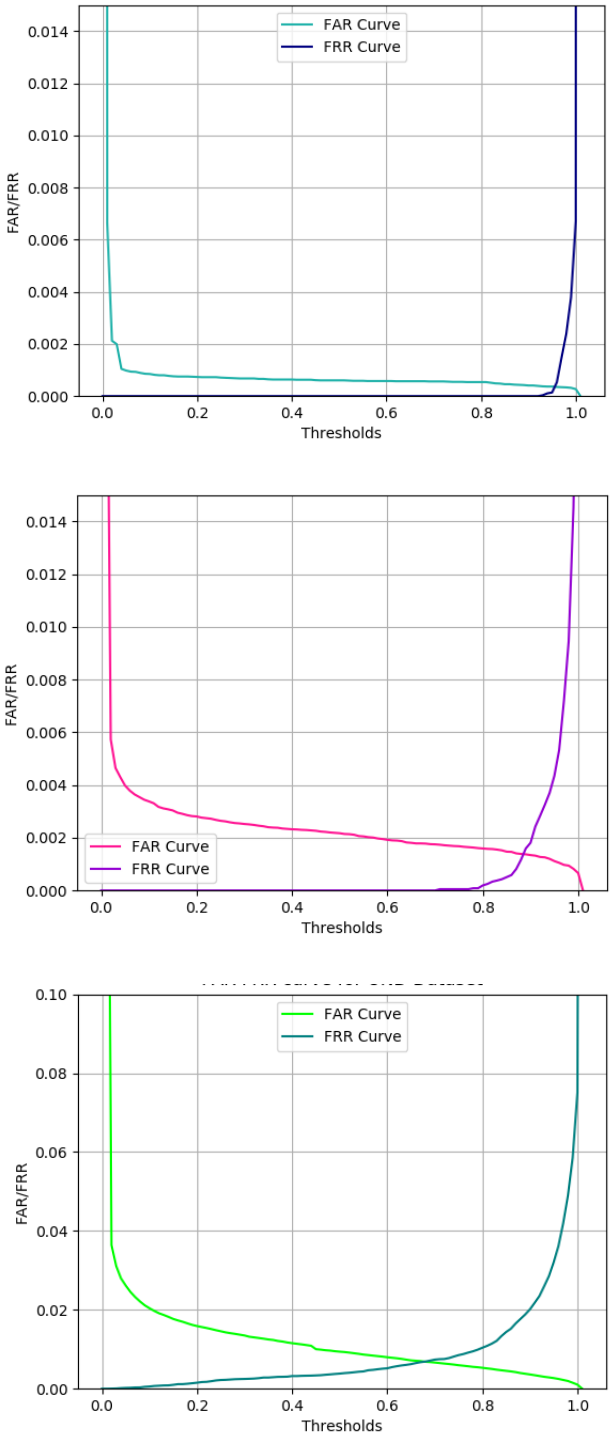



Figure 5.3: FAR-FRR curves for IITI, Bosphorus and UND databases

Table 5.3 and Table 5.4 show the results on the Proposed Model for Type I and Type II augmentations respectively where the training on the Siamese Model is done by selecting random pairs from the train features. Here, the testing is done on train-test pairs.

| Database | Training | Validation | Testing |
|---|---|---|---|
| IITI | 99.92 | 99.80 | 99.91 |
| Bosphorus | 99.71 | 99.46 | 99.66 |
| UND | 99.00 | 98.22 | 98.6 |

Table 5.3: Results for Type I Augmentation

| Database | Training | Validation | Testing |
|---|---|---|---|
| IITI | 99.83 | 99.94 | 99.21 |
| Bosphorus | 99.55 | 99.40 | 98.30 |
| UND | 99.10 | 98.61 | 96.90 |

Table 5.4: Results for Type II Augmentation

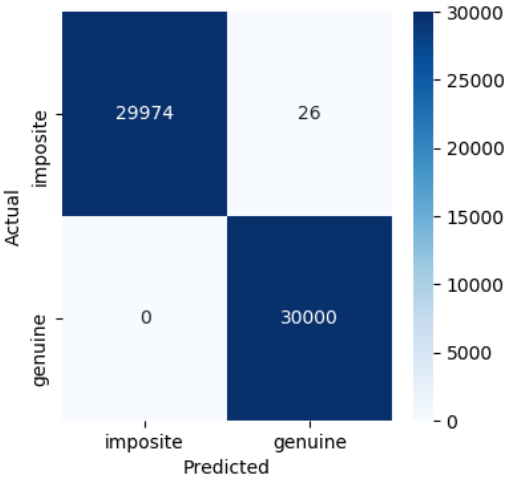
From the experiments we find that the best results are achieved for Type-I Augmentation as demonstrated in Table 5.5, Figure 5.4 and Figure 5.5.

| Database | Rank 1 Accuracy | Area Under ROC |
|---|---|---|
| IITI | 99.1 | 1.00 |
| Bosphorus | 98.3 | 0.999 |
| UND | 97.0 | 0.998 |

Table 5.5: Rank 1 accuracy and area Under ROC for Type I Augmentation

| IITI Dataset | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| Imposite | 1.00 | 1.00 | 1.00 | 30000 |
| Genuine | 1.00 | 1.00 | 1.00 | 30000 |

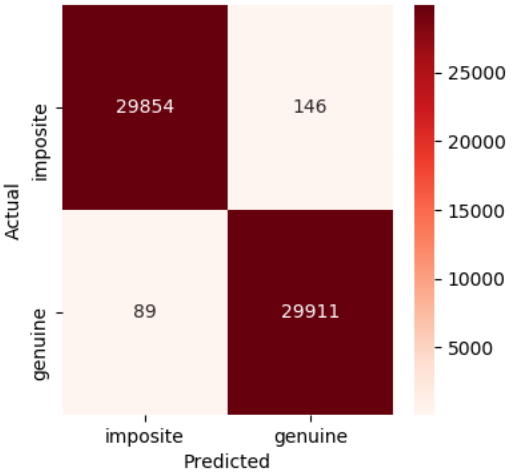
(a) Classification report for IITI database




(b) Confusion matrix for IITI database

| Bosphorus Neutral Dataset | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| Imposite | 1.00 | 1.00 | 1.00 | 30000 |
| Genuine | 1.00 | 1.00 | 1.00 | 30000 |

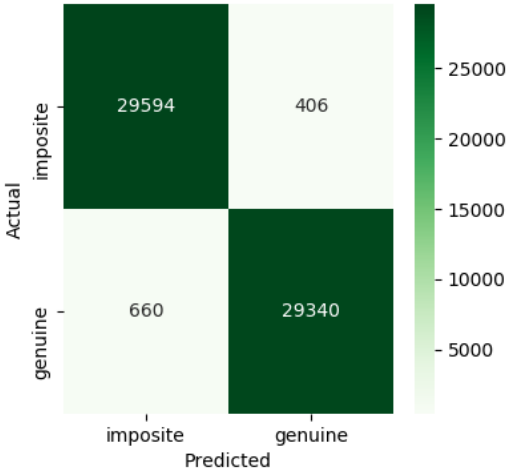
(c) Classification report for Bosphorus database




(d) Confusion matrix for Bosphorus database

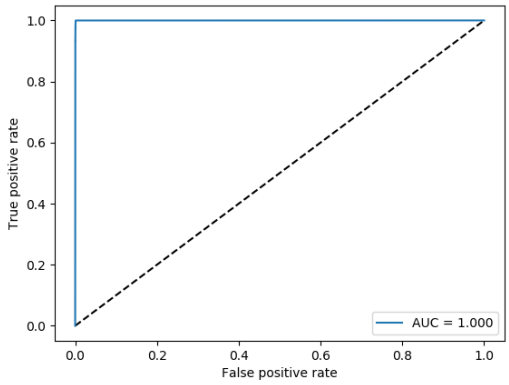
| UND Dataset | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| Imposite | 0.98 | 0.99 | 0.98 | 30000 |
| Genuine | 0.99 | 0.98 | 0.98 | 30000 |

(e) Classification report for UND database

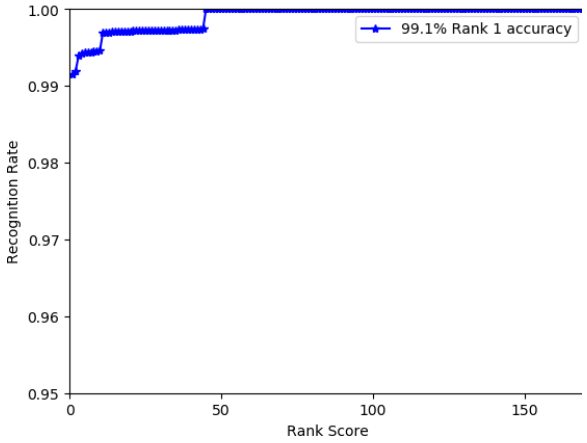



(f) Confusion matrix for UND database

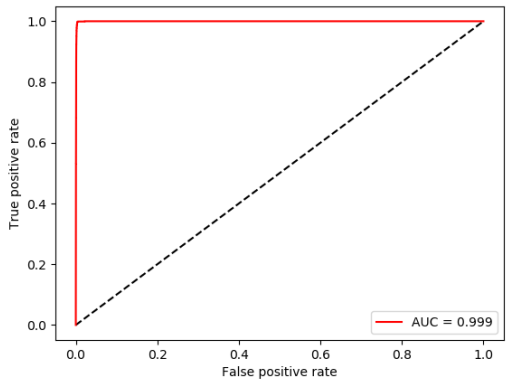
Figure 5.4: Classification report and confusion matrices



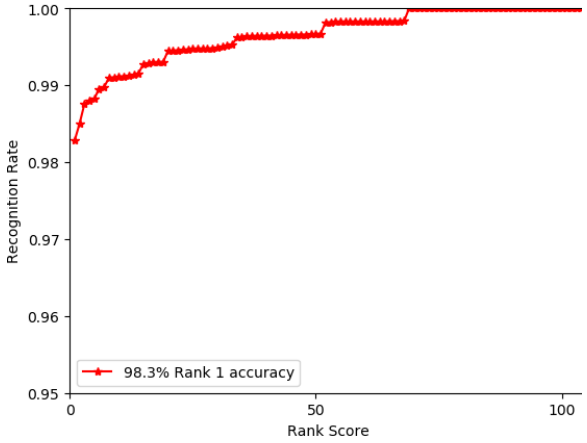

(a) ROC curve for IITI database



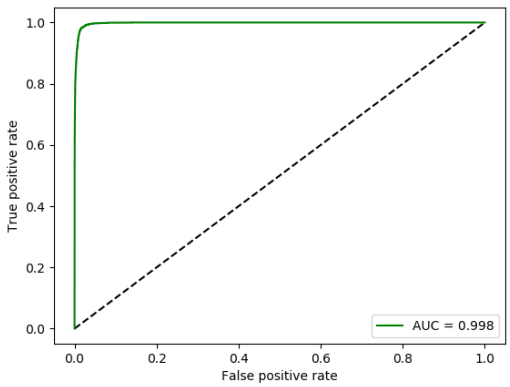

(b) CMC curve for IITI database



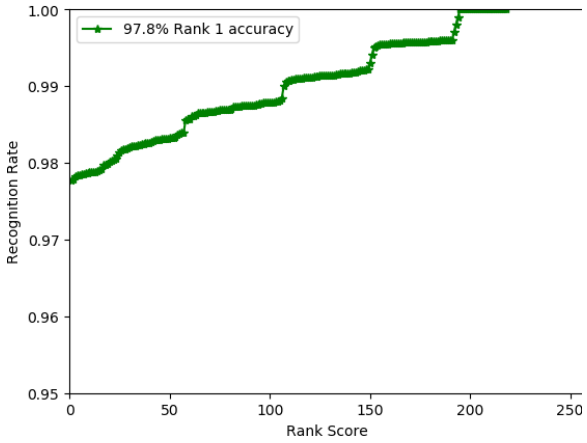

(c) ROC curve for Bosphorus database




(d) CMC curve for Bosphorus database




(e) ROC curve for UND database




(f) CMC curve for UND database

Figure 5.5: ROC and CMC curves for different databases

We also test our model using test-test pairs to find out whether our Proposed Model can predict if two test samples themselves form a genuine or imposter pair i.e. whether they belong to the same class or not. The following tables show the results for Type I augmentation.

| Database | Training | Validation | Testing |
|---|---|---|---|
| IITI | 99.95 | 99.94 | 99.95 |
| Bosphorus | 99.75 | 99.55 | 99.69 |
| UND | 98.66 | 98.96 | 98.53 |

Table 5.6: Results for Type I Augmentation (test-test pairs)

We also experiment with the different ratios of train-test split in the Siamese Network by changing it from 70-30 to 80-20 and 50-50 for Type I Augmentation.

| Database | Training | Validation | Testing |
|---|---|---|---|
| IITI | 99.93 | 99.96 | 99.91 |
| Bosphorus | 99.73 | 99.41 | 94.54 |
| UND | 98.58 | 97.89 | 97.71 |

Table 5.7: Results for train-test split: 80-20

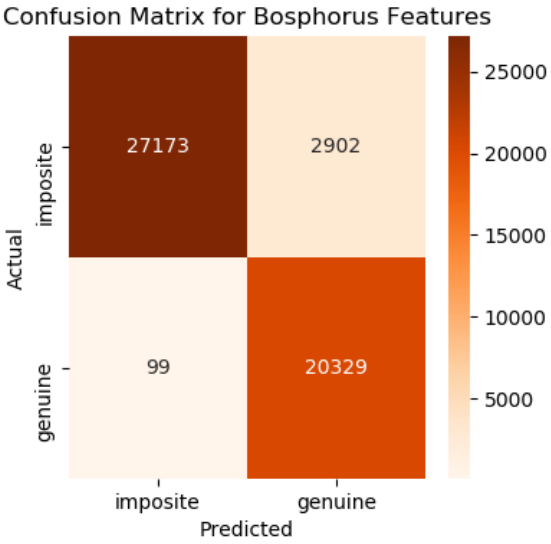
| Database | Training | Validation | Testing |
|---|---|---|---|
| IITI | 99.91 | 99.89 | 99.90 |
| Bosphorus | 99.75 | 99.67 | 99.62 |
| UND | 98.63 | 98.51 | 98.38 |

Table 5.8: Results for train-test split: 50-50

The Bosphorus database comes along with lm3 files for each 3D face scan which contains pre-existing features. On passing them through our Proposed Model, we achieve 93.54%, 92.52% and 79.17% on training, validation and testing respectively which shows that the feature extraction in our Proposed Model is much superior.

| Bosphorus Features | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| Imposite | 0.74 | 0.90 | 0.81 | 30075 |
| Genuine | 0.88 | 0.68 | 0.76 | 29925 |

(a) Classification report for Bosphorus Features




(b) Confusion matrix for Bosphorus Features

Figure 5.6: Performance of Proposed Model on Bosphorus Features

# Chapter 6

# Conclusions

In this work, a generic 3D object recognition technique is proposed that gives remarkable accuracy even on highly similar objects. In the technique, we construct the model which improves over the existing PointNet architecture by combining it with the Siamese Network with minimal preprocessing. To overcome the problem of limited 3D data samples, we also propose two new augmentation techniques where random points are selected from the point clouds of the available 3D images. We evaluate our model on three 3D face databases, namely IIT Indore, Bosphorus, and UND databases, achieving verification accuracy of 99.91%, 99.66%, and 98.60%, respectively. Our experimental results show that the best performance is achieved when Type I augmentation is used. The graphical analysis of these results also verifies that our model gives high accuracy, implying a perfect segregation between genuine and imposter pairs.

# References


[1] Charles Ruizhongtai Qi et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 77–85.

[2] D. Chetverikov, Dmitry Stepanov, and Pavel Krsek. "Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm". In: *Image and Vision Computing* 23.3 (2005), pp. 299–309.

[3] Gaile G. Gordon. "Face recognition based on depth maps and surface curvature". In: *Proceedings of SPIE - The International Society for Optical Engineering*. Vol. 1570. 1991, pp. 234–247.

[4] H. T. Tanaka, M. Ikeda, and H. Chiaki. "Curvature-based face surface recognition using spherical correlation. Principal directions for curved object recognition". In: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. 1998, pp. 372–377.

[5] Benjamin Graham. "Sparse arrays of signatures for online character recognition". In: *ArXiv* abs/1308.0371 (2013).

[6] S. Li and H. Feng. "EEG Signal Classification Method Based on Feature Priority Analysis and CNN". In: *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*. 2019, pp. 403–406.

[7] M. He, B. Li, and H. Chen. "Multi-scale 3D deep convolutional neural network for hyperspectral image classification". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 3904–3908.

[8] F. Gomez-Donoso et al. "LonchaNet: A sliced-based CNN architecture for real-time 3D object recognition". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 412–418.

[9] U. Asif, M. Bennamoun, and F. A. Sohel. "A Multi-Modal, Discriminative and Spatially Invariant CNN for RGB-D Object Labeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.9 (2018), pp. 2051–2065.

[10] W. Yun et al. "Object recognition and pose estimation for modular manipulation system: Overview and initial results". In: *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. 2017, pp. 198–201.

[11] D. O. Sales, J. Amaro, and F. S. Osório. "3D shape descriptor for objects recognition". In: *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. 2017, pp. 1–6.

[12] A. Caglayan and A. B. Can. "3D convolutional object recognition using volumetric representations of depth data". In: *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*. 2017, pp. 125–128.

[13] S. Braeger and H. Foroosh. "Curvature Augmented Deep Learning for 3D Object Recognition". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 3648–3652.

[14] A. F. Elaraby, A. Hamdy, and M. Rehan. "A Kinect-Based 3D Object Detection and Recognition System with Enhanced Depth Estimation Algorithm". In: *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 2018, pp. 247–252.

[15] S. Taertulakarn et al. "The preliminary investigation of ear recognition using hybrid technique". In: *2016 9th Biomedical Engineering International Conference (BMEiCON)*. 2016, pp. 1–4.

[16] I. I. Ganapathi et al. "Ear recognition in 3D using 2D curvilinear features". In: *IET Biometrics* 7.6 (2018), pp. 519–529.

[17] Yi Sun et al. "DeepID3: Face Recognition with Very Deep Neural Networks". In: *ArXiv* abs/1502.00873 (2015).

[18] J. Chen, V. M. Patel, and R. Chellappa. "Unconstrained face verification using deep CNN features". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2016, pp. 1–9.

[19] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. "Face Description with Local Binary Patterns: Application to Face Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), pp. 2037–2041.

[20] T. Terada, Y. Chen, and R. Kimura. "3D Facial Landmark Detection Using Deep Convolutional Neural Networks". In: *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. 2018, pp. 390–393.

[21] Hemprasad Patil, Ashwin Kothari, and K. Bhurchandi. "3-D face recognition: Features, databases, algorithms and challenges". In: *Artificial Intelligence Review* 44 (2015), pp. 393–441.

[22] Y. Lei et al. "A Two-Phase Weighted Collaborative Representation for 3D partial face recognition with single sample". In: *Pattern Recognition* 52 (2016), pp. 218–237.

[23] Huibin Li et al. "Towards 3D Face Recognition in the Real: A Registration-Free Approach Using Fine-Grained Matching of 3D Keypoint Descriptors". In: *International Journal of Computer Vision* 113.2 (2015), pp. 128–142.

[24] H. Hu et al. "2D and 3D face recognition using convolutional neural network". In: *TENCON 2017 - 2017 IEEE Region 10 Conference*. 2017, pp. 133–132.

[25] S. Zulqarnain Gilani and A. Mian. "Learning from Millions of 3D Scans for Large-Scale 3D Face Recognition". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1896–1905.

[26] D. Kim et al. "Deep 3D face identification". In: *2017 IEEE International Joint Conference on Biometrics (IJCB)*. 2017, pp. 133–142.

[27] Zhirong Wu et al. "3D ShapeNets: A deep representation for volumetric shapes". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1912–1920.

[28] Daniel Maturana and Sebastian A. Scherer. "VoxNet: A 3D Convolutional Neural Network for real-time object recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), pp. 922–928.

[29] Charles Ruizhongtai Qi et al. "Volumetric and Multi-view CNNs for Object Classification on 3D Data". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 5648–5656.

[30] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. "The wave kernel signature: A quantum mechanical approach to shape analysis". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (2011), pp. 1626–1633.

[31] Michael M. Bronstein and Iasonas Kokkinos. "Scale-invariant heat kernel signatures for non-rigid shape recognition". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), pp. 1704–1711.

[32] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. "Fast Point Feature Histograms (FPFH) for 3D registration". In: *2009 IEEE International Conference on Robotics and Automation* (2009), pp. 3212–3217.

[33] Kan Guo, Dongqing Zou, and Xiaowu Chen. "3D Mesh Labeling via Deep Convolutional Neural Networks". In: *ACM Transactions on Graphics* 35.1 (2015), 3:1–3:12.

[34] Anastasia Ioannidou et al. "Deep Learning Advances in Computer Vision with 3D Data: A Survey". In: *ACM Computing Surveys* 50.2 (2017), 20:1–20:38.

[35] H. Wu et al. "Face recognition based on convolution siamese networks". In: *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. 2017, pp. 1–5.

[36] W. Hayale, P. Negi, and M. Mahoor. "Facial Expression Recognition Using Deep Siamese Neural Networks with a Supervised Loss function". In: *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*. 2019, pp. 1–7.

[37] J. C. Joshi et al. "Scanned to Digital Face Images Matching With Siamese Network". In: *2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*. 2018, pp. 1–6.

[38] Arman Savran et al. "Bosphorus Database for 3D Face Analysis". In: *Workshop on Biometrics and Identity Management*. 2008, pp. 47–56.

[39] Kyong I. Chang, Kevin W. Bowyer, and Patrick J. Flynn. "Face recognition using 2D and 3D facial data". In: *ACM Workshop on Multimodal User Authentication*. 2003, pp. 25–32.

[40] Patrick J. Flynn, Kevin W. Bowyer, and P. Jonathon Phillips. "Assessment of Time Dependency in Face Recognition: An Initial Study". In: *Audio- and Video-Based Biometric Person Authentication*. Ed. by Josef Kittler and Mark S. Nixon. 2003, pp. 44–51.