Facial Recognition

CS 613 Machine Learning

1st Mahima Modi
College of Computing and Informatics
Drexel University
Philadelphia, PA, USA
mjm838@drexel.edu

2nd Brinda Kulkarni
College of Computing and Informatics
Drexel University
Philadelphia, PA, USA
bk644@drexel.edu

3rd Raghad Aseeri
College of Computing and Informatics
Drexel University
Philadelphia, PA, USA
ra869@drexel.edu

***Abstract*—Facial recognition is gaining a lot of momentum and has a numerous applications in the security and biometrics areas. The facial images are extracted from a set of videos and classified using the models proposed, which include a Convolution Neural Network with 5 convolution layers and the well-known VGGResnet50 model, which fine-tunes the feature vectors to a large extent, along with the classifiers SVM, KNN, and Softmax. We may compare and analyze the performance of each model in the Experiments and Analysis section.**

***Index Terms*—CNN,VGGResnet50,SVM,KNN,Softmax**

I. Background

Machine learning and deep learning techniques have advanced significantly in recent years. It is widely used in our daily lives in various fields such as health care, finance, social networking. The Biometrics system is one such application, in which the network gets familiar with the many patterns supplied into it and identifies similar patterns. It uses fingerprint, retina, and iris recognition, as well as hand/palm, voice, signature, and face identification. In this paper, we demonstrate Facial Recognition which has various applications like Smart Advertising, disease diagnosis, identity validation at ATMs to prevent crime, assisting the blind in recognizing the emotions on others' faces, and finding missing people.Facial Recognition is an important tool for human computer interaction. Here we discuss how the training images are classified with CNN as well as the Resnet50 model with SVM,KNN and Softmax classifier. We are making use of images of 21 classes to train the models. The performance of every model is compared and conclusion is drawn. We have built a Convolution layered network from scratch which classifies the images using Softmax Classifier. Further we have employed the VGGResnet50 to improve the extraction of the feature vectors and classified them using SVM,KNN and Softmax model. The remaining sections of this paper are structured as follows. The second section elaborates the Related work on the paper. In Section 3,the CNN, VGGResnet50, SVM, KNN and Softmax models are explained.Section 4 displays a sample of the dataset.Section 5 discusses the Proposed Algorithm where the working of the models used is explained. The results of the experiment are presented in Section 6. Finally, in section 6, we talk about the conclusion.

Identify applicable funding agency here. If none, delete this.

II. Related Work

A. Image Net Classification with Deep Convolutional Neural Networks

Convolution neural networks feature fewer connections than feedforward neural networks, yet they perform better than ANNs even with fewer parameters. The activation functions, when converted to ReLU, speed up error convergence and avoid saturation. Other methods, such as local response normalization and overlapping pooling, reduce the error proportion dramatically.

B. An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification

The paper builds a CNN-SVM model on the MNIST dataset and compares its accuracy with the CNN-Softmax model for the same. The image classification architecture is recreated in this research by combining a convolutional neural network (CNN) and a linear SVM. The CNN utilized in this work is a simple 2-Convolutional Layer with Max Pooling model. The goal of SVM is to determine the best hyperplane to differentiate classes in a dataset. Because SVM computes weights using the Manhattan distance, which has a typical hinge loss, the Eucledian distance is employed instead to produce more stable results.

C. Face Recognition Based on Convolution Neural Network

This paper discusses how Convolutional Neural Networks (CNN), may be used to extract features and classify them in a single architecture for face recognition. A CNN model with a SoftMax Classifier is proposed in this study to recognize faces from the dataset. CNN is a large-scale classification algorithm with numerous deep layers, each having a tiny convolution layer with a different number of kernels, followed by max pooling.

D. Comparative Study of Image Classification using Machine Learning Algorithms

From this paper we will use CNN with KNN classifier. After that, we are planning to use similarity methods like euclidean distance, and cosine similarity. The k-nearest neighbor algorithm estimates the predictors within each class by looks for in the observation to find the nearest points to predictor points and response values to those nearest points, and then it classifies an observation by estimating the posterior probability for each class and expected classification cost; Let y is the predicted classification, K is the number of classes, P Ù (k—x) is the posterior probability of class k for observation x and C(y—k) is the cost of classifying an observation as y when its true class is k.

III. Methodology

A. CNN

It is a type of deep learning artificial neural network that is used for image classification, identification, segmentation, and detection of objects. It is based on the concept of layers, with the first layer being the image input, followed by the convolutions and pooling layers. The images are distinguished based on the weights and biases assigned to their features.

Convolution Layer - CNN's primary component is the Convolutional layer. It uses filters to extract feature vectors from images represented as matrices, which is accomplished by multiplying the stride of the filter by the chunk of the filter's size matrix as it moves forward. We ultimately have a two-dimensional image that serves as an input for the next layer.

Pooling Layer - The image is partitioned into rectangles in this layer, and one feature vector is selected from each rectangle. For the dimensionality reduction of the feature vectors, two types of pooling are used: max pooling and average pooling. This lowers the overall calculation cost and assures that the layer converges more quickly.

Fully Connected Layer - This layer consists of a feed forward network connected by neurons. It takes the flattened images as input and executes mathematical operations on them every time iteration occurs across a number of epochs. Finally, the high-level features were extracted, which are then sent into the classification layer.

Rectified Linear Activation Unit(RELU) - This has been the most effective activation function since it converges much faster and is simple to train. It converts the input vectors by selecting the maximum value and returning zero in the case of negative values.

DropOut- DropOut is a simple strategy to reduce overfitting in neural networks, which can occur when training data is large. It can be used with the input layer to eliminate particular features or with the hidden layer to remove neurons to support regularization.

B. Resnet

The Residual Network was designed to fix the issue of accuracy decrease as the number of training layers in a deep convolutional neural network increased. It has 48 convolutional layers, 1 max pool layer, and 1 average pool layer in its architecture, with a total operation of 3.8 x 109 floating points. The above design is utilized to create Residual Blocks, which are then employed with the Neural Network. It ensures that the accuracy of the upper layers is not less than that of the lower levels since the core of the block skips the connections by guaranteeing that the gradient passes through alternately. In the realm of computer vision, it has a high usability.

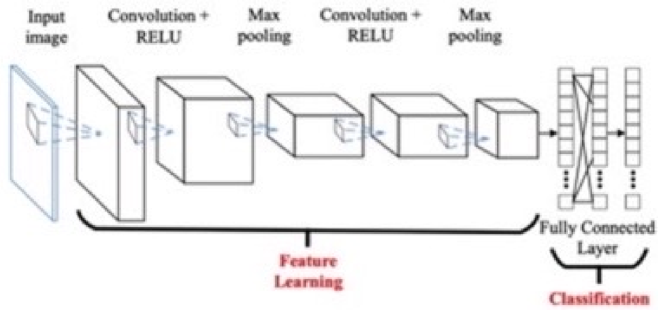


Fig. 1. CNN Layered Architecture

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 2. Resnet50 Architecture

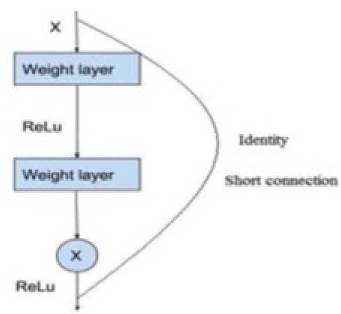


Fig. 3. ResBlock

C. SVM

- SVM algorithm is one of the ways of machine learning where the model is trained by input data and expected output data. SVM is a member of the kernel-based classifier family, which implicitly maps input features into a higher-dimensional feature space using a kernel function that measures the distance between feature points in the mapped space, allowing SVM to outperform traditional linear classification methods.
- Multi-label SVM uses the one-versus4 method to combine predictions of multiple hinary SVM classifiers. Let 2, = (XI, yl)¿ ..., (X , ,Y ,), where x,(xi E R") denotes

a n-dimensional feature vector, its p-dimensional label vector is yi(yi E Y") which consists of -I, +I, where +I means the data belongs to this class and -I mean the data does not. p is the size of label set. The j-th component of y; corresponds to the output of j-th binary SVM. Given 2). each class is seperated from all other classes by a binary SVM classifier. Then p binary SVM classifiers are obtained in total. For i-th (i = 1, ...p) classifier, it can he written as (w;, x) + b; = 0. w: and 6: can be found by: minirnize: qw;) = ;1/w,1/2+ CC,"=,[lp s.ubject to : yk ((wi, xk) + bi) 2 1 - [k, k = 1,. . . , m where (2 0) is the k-th slack variable and C is the parameter controlling the trade-off hetween function complexity and training error. For each binary SVM classifier fi, a threshold t, will he set. This will be discussed in the next section.

$$\min_{w} \frac{1}{2} \sum_{i=1}^{n} w_i^2 + C \sum_{j=1}^{m} max(0, 1 - t_j.y_j)$$

The hinge loss is a specific type of cost function that incorporates a margin or distance from the classification boundary into the cost calculation.

D. KNN

is a method for classifying objects based on closest training examples in the feature space. k-nearest neighbor algorithm is among the simplest of all machine learning algorithms. Training process for this algorithm only consists of storing feature vectors and labels of the training images. In the classification process, the unlabelled query point is simply assigned to the label of its k nearest neighbors.

- Since we are using K-nearest Neighbours obvious question comes is what K is appropriate for the clustering dataset.
- K=0 means clustering only samples and its augmentations together.
- $$K \geq 1$$
  captures more of the cluster's variance and has chances of increasing noise i.e. not all samples and their neighbors belong to the same cluster.

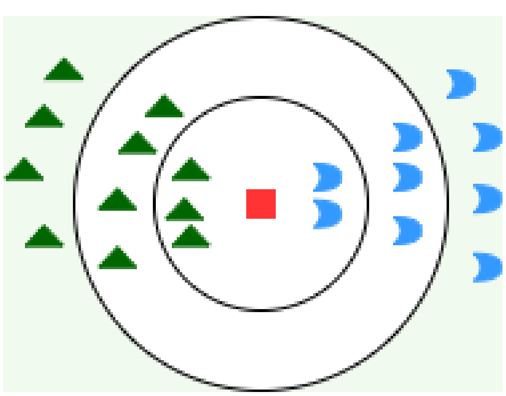


Fig. 4. Example of a figure caption.

E. Softmax

Softmax function interprets the vector values fed into it into probabilities which sums up to 1. It is usually used for multi class classification when the input has mutually exclusive classes. Its working is similar to logistic regression model. It is used at the final layer of the neural network as it helps in the normalization of the values.

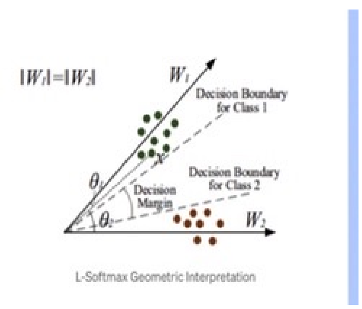


Fig. 5. Collection of images per class.

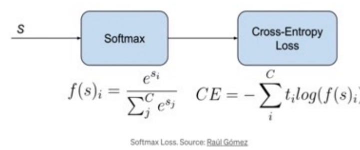


Fig. 6. Collection of images per class.

IV. DATASET COLLECTION

To train the model, we attempted to gather data in the form of images/videos on our own.We collected videos of people which shows their face in different angles and expressions.

Fig. 7. Collection of images per class.

We extracted the images from each frame of the movie using OpenCV. The faces in each frame are detected using MTCNN to remove background learning. Age, skin color, brightness, and face expression are among the dataset's variations.

The dataset contains 21 unique individuals. There are 6127 photos in Training, divided into 21 categories. There are 698 photos in testing, divided into 21 categories (nearly 10 percent of dataset). Data augmentation was performed Rescale the images from (0,255) to (0,1) and augment the images using techniques such as Brightness, 60-degree rotation, and horizontal flipping.

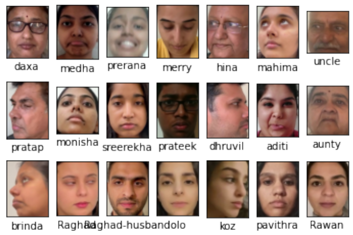


Fig. 8. Collection of images per class.

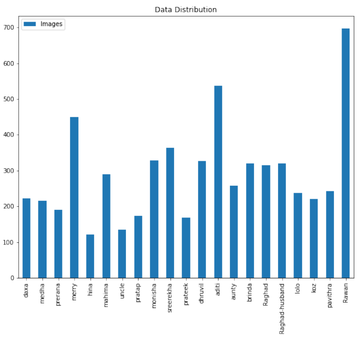


Fig. 9. Dataset Distribution of images per class.

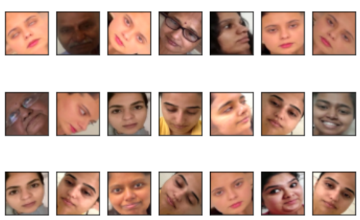


Fig. 10. Data Augmentation.

V. Proposed Algorithms

A. Method 1: VGG-ResNet50 with KNN

The preprocessed frames of the images of size 244x244x3 are fed into RESNET50 block.

The output obtained from the RESNET50 block is fed into the flattening layer.

The output obtained from the flattening layer of size 1x1x2048 is fed into the the first dense layer.

The output obtained from the first dense layer of size 1x1x1024 is fed into the second dense layer.

The output obtained from the second dense layer of size 1x1x512 into are fed into the KNN classification layer.

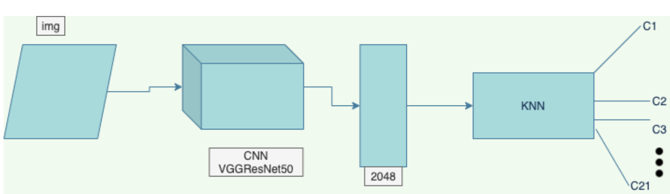


Fig. 11. Collection of images per class.

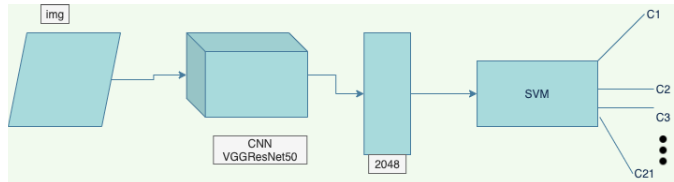


Fig. 12. Example of SVM.

B. Method 2: VGG-ResNet50 with SVM

The preprocessed frames of the images of size 244x244x3 are fed into RESNET50 block.

The output obtained from the RESNET50 block is fed into the flattening layer.

The output obtained from the flattening layer of size 1x1x2048 is fed into the the first dense layer.

The output obtained from the first dense layer of size 1x1x1024 is fed into the second dense layer.

The output obtained from the second dense layer of size 1x1x512 into are fed into the SVM classification layer.

C. Method 3: Three Layer CNN with Softmax

Feature Extraction Model- Build CNN model with 8 layers namely 3 convolution with RELU activation, 2 maxpooling, fully computed layer with 2 dense layers and 1 flattening layer. The preprocessed frames of the images of size 244x244x3 are fed into the CNN Model.

Dropout is used nullify the hidden neurons and get the dimensions 1x1x512.

The feature vectors of size 1x1x512 are fed it into the classification layer which is softmax in this case.

21 classes of images are obtained and the predicted value is the maximum of all the values among all the classes for that image.

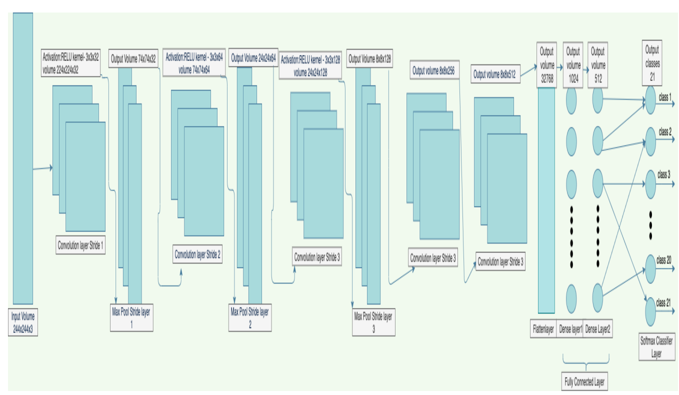


Fig. 13. CNN Layered Architecture

D. Method 4: VGG-ResNet50 with Softmax

The preprocessed frames of the images of size 244x244x3 are fed into RESNET50 block. The output obtained from the RESNET50 block is fed into the flattening layer.

The output obtained from the flattening layer of size 1x1x2048 is fed into the the first dense layer.

The output obtained from the first dense layer of size 1x1x1024 is fed into the second dense layer.

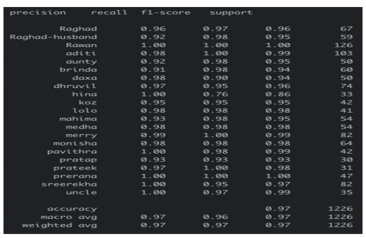
The output obtained from the second dense layer of size 1x1x512 into are fed into the Softmax classification layer.

VI. Experiments and Results

A. KNN

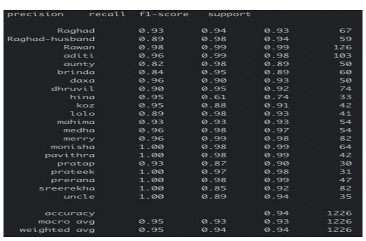
The above table shows results for k-nearest neighbor classification model. Here, we can see that the test accuracy is 0.97 and the validation accuracy is 0.97 when we take k=1. This is higher than the accuracy that was computed for k=3 and k=5 and hence is the best option in this case. We can also observe that it has high precision, recall and f1- score compared to k=3 and k=5. Results

KNN	precision	recall	f1-score	support
k=5	0.93	0.92	0.92	1226
k=3	0.95	0.93	0.93	1226
k=1	0.97	0.96	0.97	1226



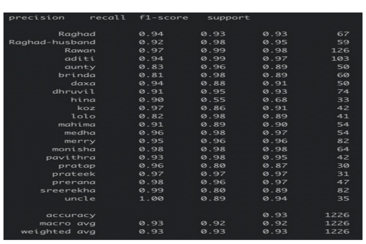
	precision	recall	f1-score	support
Raghad	0.96	0.97	0.96	67
Raghad-husband	0.92	0.98	0.95	59
Rawan	1.00	1.00	1.00	126
aditi	0.98	1.00	0.99	103
aunty	0.92	0.98	0.95	50
brinda	0.91	0.98	0.94	60
daxa	0.98	0.90	0.94	50
dhruvil	0.97	0.95	0.96	74
hina	1.00	0.76	0.86	33
koz	0.95	0.95	0.95	42
lolo	0.98	0.98	0.98	41
mahima	0.93	0.98	0.95	54
medha	0.98	0.98	0.98	54
merry	0.99	1.00	0.99	82
monisha	0.98	0.98	0.98	64
pavithra	1.00	0.98	0.99	42
pratap	0.93	0.93	0.93	30
prateek	0.97	1.00	0.98	31
prerana	1.00	1.00	1.00	47
sreerekha	1.00	0.95	0.97	82
uncle	1.00	0.97	0.99	35
accuracy			0.97	1226
macro avg	0.97	0.96	0.97	1226
weighted avg	0.97	0.97	0.97	1226

Fig. 14. KNN with n=1



	precision	recall	f1-score	support
Raghad	0.93	0.94	0.93	67
Raghad-husband	0.89	0.98	0.94	59
Rawan	0.98	0.99	0.99	126
aditi	0.96	0.99	0.98	103
aunty	0.82	0.98	0.89	50
brinda	0.84	0.95	0.89	60
daxa	0.96	0.90	0.93	50
dhruvil	0.90	0.95	0.92	74
hina	0.95	0.61	0.74	33
koz	0.95	0.88	0.91	42
lolo	0.89	0.98	0.93	41
mahima	0.93	0.93	0.93	54
medha	0.96	0.98	0.97	54
merry	0.96	0.99	0.98	82
monisha	1.00	0.98	0.99	64
pavithra	1.00	0.98	0.99	42
pratap	0.93	0.87	0.90	30
prateek	1.00	0.97	0.98	31
prerana	1.00	0.98	0.99	47
sreerekha	1.00	0.85	0.92	82
uncle	1.00	0.89	0.94	35
accuracy			0.94	1226
macro avg	0.95	0.93	0.93	1226
weighted avg	0.95	0.94	0.94	1226

Fig. 15. KNN with n=3



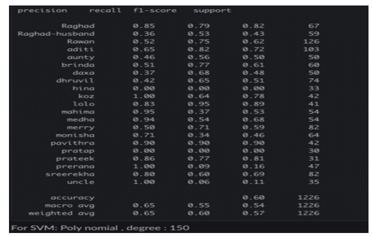
	precision	recall	f1-score	support
Raghad	0.94	0.93	0.93	67
Raghad-husband	0.92	0.98	0.95	59
Rawan	0.97	0.99	0.98	126
aditi	0.94	0.99	0.97	103
aunty	0.83	0.96	0.89	50
brinda	0.81	0.98	0.89	60
daxa	0.94	0.88	0.91	50
dhruvil	0.91	0.95	0.93	74
hina	0.90	0.55	0.68	33
koz	0.97	0.86	0.91	42
lolo	0.82	0.98	0.89	41
mahima	0.91	0.89	0.90	54
medha	0.96	0.98	0.97	54
merry	0.95	0.96	0.96	82
monisha	0.98	0.98	0.98	64
pavithra	0.93	0.98	0.95	42
pratap	0.96	0.80	0.87	30
prateek	0.97	0.97	0.97	31
prerana	0.98	0.96	0.97	47
sreerekha	0.99	0.80	0.89	82
uncle	1.00	0.89	0.94	35
accuracy			0.93	1226
macro avg	0.93	0.92	0.92	1226
weighted avg	0.93	0.93	0.93	1226

Fig. 16. KNN with n=5

B. SVM

From the above results, we can observe the test accuracy and validation accuracy of the different methods that we used. It is very evident from the results that SVM-Polynomial-(degree 100) has the least accuracy and SVM-RBF-(gamma 50) has the highest accuracy. Hence, we were able to conclude that SVM-RBF-(gamma 50) is the best model in this case. We also calculated the precision, recall, f1 score and support for each of the models. Again, it is evident that SVM-RBF-(gamma 50) gives the best results.

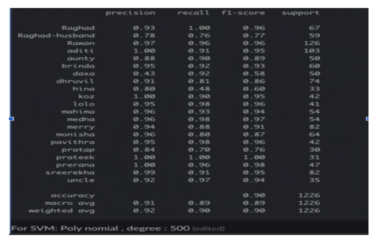
Models	precision	recall	f1-score	support
SVM-Linear	0.42	0.42	0.39	1226
SVM-Polynomial-(degree: 100)	0.41	0.40	0.37	698
SVM-Polynomial-(degree: 250)	0.81	0.78	0.78	1226
SVM-Polynomial-(degree: 500)	0.91	0.89	0.89	1226
SVM-RBF-(gamma: 10)	0.91	0.87	0.88	698
SVM-RBF-(gamma: 50)	0.98	0.97	0.98	698



	precision	recall	f1-score	support
Raghad	0.85	0.79	0.82	67
Raghad-husband	0.36	0.53	0.43	59
Rawan	0.52	0.75	0.62	126
aditi	0.65	0.82	0.72	103
aunty	0.46	0.56	0.50	50
brinda	0.51	0.77	0.61	60
daxa	0.37	0.68	0.48	50
dhruvil	0.42	0.65	0.51	74
hina	0.00	0.00	0.00	33
koz	1.00	0.64	0.78	42
lolo	0.83	0.95	0.89	41
mahima	0.95	0.37	0.53	54
medha	0.94	0.54	0.68	54
merry	0.50	0.71	0.59	82
monisha	0.71	0.34	0.46	64
pavithra	0.90	0.90	0.90	42
pratap	0.00	0.00	0.00	30
prateek	0.86	0.77	0.81	31
prerana	1.00	0.09	0.16	47
sreerekha	0.80	0.60	0.69	82
uncle	1.00	0.06	0.11	35
accuracy			0.60	1226
macro avg	0.65	0.55	0.54	1226
weighted avg	0.65	0.60	0.57	1226

For SVM: Poly nomial , degree : 150

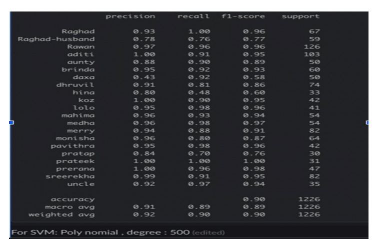
Fig. 17. Polynomial SVM with degree 150



	precision	recall	f1-score	support
Raghad	0.93	1.00	0.96	67
Raghad-husband	0.78	0.76	0.77	59
Rawan	0.97	0.96	0.96	126
aditi	1.00	0.91	0.95	103
aunty	0.88	0.90	0.89	50
brinda	0.95	0.92	0.93	60
daxa	0.43	0.92	0.58	50
dhruvil	0.91	0.81	0.86	74
hina	0.80	0.48	0.60	33
koz	1.00	0.90	0.95	42
lolo	0.95	0.98	0.96	41
mahima	0.96	0.93	0.94	54
medha	0.96	0.98	0.97	54
merry	0.94	0.88	0.91	82
monisha	0.96	0.80	0.87	64
pavithra	0.95	0.98	0.96	42
pratap	0.84	0.70	0.76	30
prateek	1.00	1.00	1.00	31
prerana	1.00	0.96	0.98	47
sreerekha	0.99	0.91	0.95	82
uncle	0.92	0.97	0.94	35
accuracy			0.90	1226
macro avg	0.91	0.89	0.89	1226
weighted avg	0.92	0.90	0.90	1226

For SVM: Poly nomial , degree : 500 (edited)

Fig. 18. Polynomial SVM with degree 500



	precision	recall	f1-score	support
Raghad	0.93	1.00	0.96	67
Raghad-husband	0.78	0.76	0.77	59
Rawan	0.97	0.96	0.96	126
aditi	1.00	0.91	0.95	103
aunty	0.88	0.90	0.89	50
brinda	0.95	0.92	0.93	60
daxa	0.43	0.92	0.58	50
dhruvil	0.91	0.81	0.86	74
hina	0.80	0.48	0.60	33
koz	1.00	0.90	0.95	42
lolo	0.95	0.98	0.96	41
mahima	0.96	0.93	0.94	54
medha	0.96	0.98	0.97	54
merry	0.94	0.88	0.91	82
monisha	0.96	0.80	0.87	64
pavithra	0.95	0.98	0.96	42
pratap	0.84	0.70	0.76	30
prateek	1.00	1.00	1.00	31
prerana	1.00	0.96	0.98	47
sreerekha	0.99	0.91	0.95	82
uncle	0.92	0.97	0.94	35
accuracy			0.90	1226
macro avg	0.91	0.89	0.89	1226
weighted avg	0.92	0.90	0.90	1226

For SVM: Poly nomial , degree : 500 (edited)

Fig. 19. Polynomial SVM with degree 1000

VII. Conclusions

We may conclude that the VVGResnet50 model with Softmax classifier has the best performance for the given dataset of images, with a 100 percent accuracy rate and the ability to accurately classify five out of six photos when evaluated with on-ground images. In comparison with the other models of VGGResnet50 model with SVM, KNN which have poor onground performance.If the dataset contains images with variations, the model will be trained in a better environment and will be able to predict values that are not beyond its reach.

	precision	recall	f1-score	support
Raghad	0.86	0.91	0.89	35
Raghad-husband	0.83	1.00	0.91	35
Rawan	0.91	0.97	0.94	80
aditi	0.97	1.00	0.98	60
aunty	0.80	0.93	0.86	30
brinda	0.74	0.89	0.81	35
daxa	0.85	0.81	0.83	27
dhruvil	0.88	0.79	0.83	38
hina	1.00	0.33	0.50	15
koz	1.00	0.84	0.91	25
lolo	0.89	0.92	0.91	26
mahima	0.89	0.76	0.82	33
medha	0.96	1.00	0.98	24
merry	0.88	0.96	0.92	51
monisha	0.94	0.94	0.94	36
pavithra	0.96	0.93	0.94	27
pratap	0.84	0.73	0.78	22
prateek	1.00	1.00	1.00	19
prerana	1.00	0.87	0.93	23
sreerekha	0.90	0.84	0.87	45
uncle	0.90	0.75	0.82	12
accuracy			0.90	698
macro avg	0.91	0.87	0.88	698
weighted avg	0.90	0.90	0.89	698

Fig. 20. Rbf SVM with gamma 10

	precision	recall	f1-score	support
Raghad	1.00	0.97	0.99	35
Raghad-husband	1.00	1.00	1.00	35
Rawan	0.99	1.00	0.99	80
aditi	1.00	1.00	1.00	60
aunty	0.88	1.00	0.94	30
brinda	0.89	0.97	0.93	35
daxa	1.00	1.00	1.00	27
dhruvil	1.00	0.95	0.97	38
hina	0.93	0.93	0.93	15
koz	1.00	0.96	0.98	25
lolo	0.96	0.96	0.96	26
mahima	1.00	1.00	1.00	33
medha	0.96	1.00	0.98	24
merry	1.00	0.98	0.99	51
monisha	0.97	0.97	0.97	36
pavithra	1.00	1.00	1.00	27
pratap	1.00	0.95	0.98	22
prateek	1.00	1.00	1.00	19
prerana	1.00	0.96	0.98	23
sreerekha	0.98	0.98	0.98	45
uncle	1.00	0.83	0.91	12
accuracy			0.98	698
macro avg	0.98	0.97	0.98	698
weighted avg	0.98	0.98	0.98	698

Fig. 21. Rbf SVM with gamma 50

VIII. Future Work/Extensions

We should be able to detect the faces irrespective of the vulnerabilities like the background of the image,and other obstacles like presence of glasses, beard, mask.We could include more augmentation techniques to enhance the model's accuracy. We could implement Siamese Network for face recognition.

References

[1] Agarap, A. F. (2019, February 7). An architecture combining convolutional neural network (CNN) and Support Vector Machine (SVM) for Image Classification. arXiv.org.

[2] J. Xiang and G. Zhu, "Joint Face Detection and Facial Expression Recognition with MTCNN," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 2017, pp. 424-427, doi: 10.1109/ICISCE.2017.95.

[3] ImageNet classification with deep convolutional ... - neurips. (n.d.).

[4] Comparative Study of Image Classification using Machine Learning Algorithms. (n.d.). Retrieved November 23, 202.

[5] S. Lawrence, C. L. Giles, Ah Chung Tsoi and A. D. Back, "Face recognition: a convolutional neural-network approach," in IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 98-113, Jan. 1997, doi: 10.1109/72.554195.

[6] Agarap, A. F. (2019, February 7). An architecture combining convolutional neural network (CNN) and Support Vector Machine (SVM) for Image Classification. arXiv.org. https://arxiv.org/abs/1712.03541

[7] x. Fu, J. Lu, X. Zhang, X. Yang and I. Unwala, "Intelligent In-Vehicle Safety and Security Monitoring System with Face Recognition," 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2019, pp. 225-229, doi: 10.1109/CSE/EUC.2019.00050.

[8] K. Yan, S. Huang, Y. Song, W. Liu and N. Fan, "Face recognition based on convolution neural network," 2017 36th Chinese Control Conference (CCC), 2017, pp. 4077-4081, doi: 10.23919/ChiCC.2017.8027997.

[9] A. Peña, A. Morales, I. Serna, J. Fierrez and A. Lapedriza, "Facial Expressions as a Vulnerability in Face Recognition," 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 2988-2992, doi: 10.1109/ICIP42928.2021.9506444.