Project 1.

Is that you? Metric Learning Approaches for Face Identification

Team Ikshana

Ananya Arun (20171019)
Samhita Kanaparthy (2018121005)
Sravani Kaza (20171189)
Thejasvi Konduru (20171134)

LFW Dataset


Elisabeth Schumacher, 1 Elisabeth Schumacher, 2


Debra Messing, 1 Debra Messing, 2

Labeled Faces in the Wild Dataset

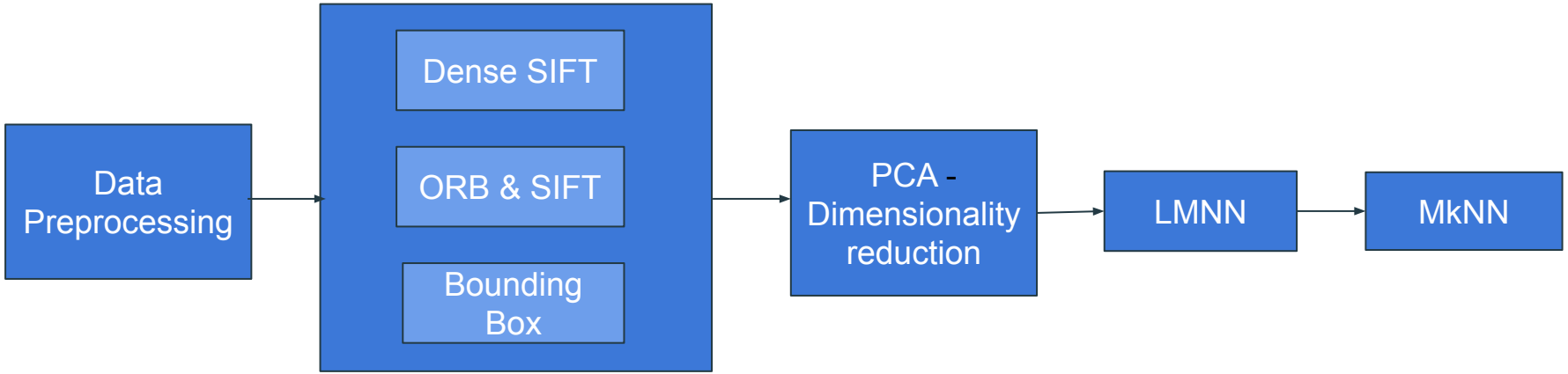
- 13233 images
- 5749 people
- 1680 people with two or more images

The paper runs the experiment in two different settings

- Restricted - fixed set of positive and negative image pairs given
- Unrestricted - faces labeled by their identity

In our project, we limited our experiments to the Unrestricted settings.

Pipeline


Data Preprocessing
Dense SIFT
ORB & SIFT
Bounding Box
PCA - Dimensionality reduction
LMNN
MkNN

Data Preprocessing & Dimensionality Reduction

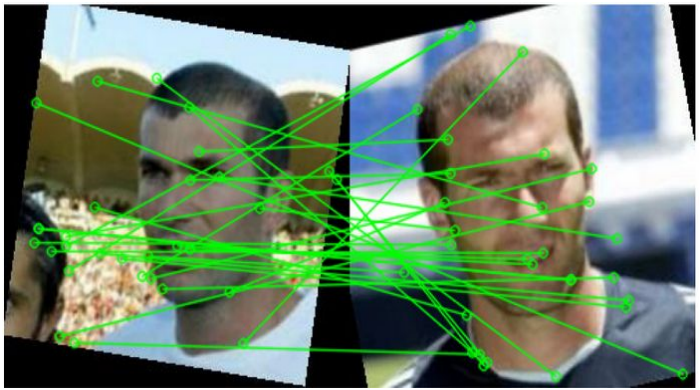
- N - no.of images exist for each person in the dataset.
- For our experiments, we consider the images of only those persons with N>3 for better results.
- We do feature extraction on this refined dataset.
- Then dimensionality reduction PCA is performed on these extracted features.

Feature Extraction

In our experiments, we used the following feature extractors

- **Scale-invariant feature transform (SIFT)**
- **Oriented FAST and Rotated BRIEF (ORB)**

ORB



SIFT

Large Margin Nearest Neighbour Metrics(LMNN)

Mahalanobis distance:

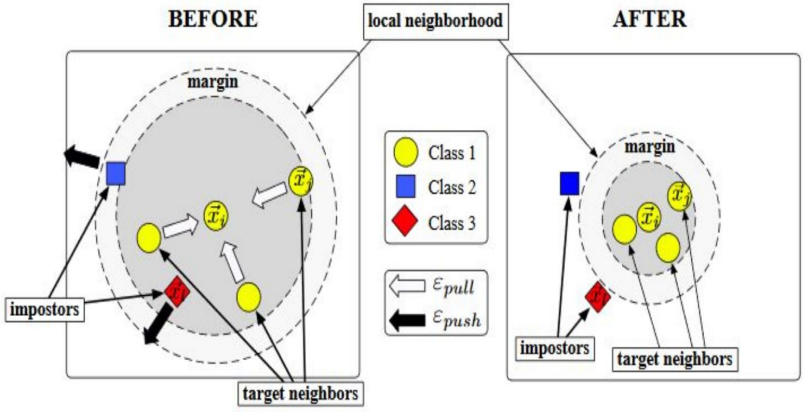
$$d(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j)$$

Imposters:

An impostor of a data point x_i is another data point x_j if both are not in same class ($y_i != y_j$), but is a nearest neighbour of x_i

LMNN (Large Margin Nearest Neighbour Metrics)

Objective: To Maintain a large (finite) distance between imposters and the perimeters established by target neighbours. This robustness criterion gives rise to the name of approach: large margin nearest neighbour (LMNN) classification.



Large Margin Nearest Neighbour Metrics (LMNN)

- The intuition is that for each data point we need a metric which makes the k nearest neighbours of its own class – target neighbours – closer than points from other classes.
- The objective has two term, one minimises the distances between target neighbours
- The other is a hinge-loss that encourages target neighbours to be at least one distance unit closer than points from other classes.

Large Margin Nearest Neighbour Metrics (LMNN)

- The first term of the loss function (O1), minimizes the average distance between instances and their target neighbors.

$$O1 = \sum_{i,j \in N_i} d(x_i, x_j)$$

- The gradient of this term generates a pulling force that attracts target neighbors in the linearly transformed input space.

Large Margin Nearest Neighbour Metrics (LMNN)

- The second term in the loss function (O2) minimizes x_i that are less than one unit further away than target neighbors x_j, hence pushing the samples of other class away.

$$O2 = \sum_{i,j \in N_i, l, y_l \neq y_i} [d(x_i, x_j) + 1 - d(x_i, x_l)]_+$$

- Here x_l denotes imposters of x_i and x_j are neighbours of x_i.

Large Margin Nearest Neighbour Metrics (LMNN)

- Now we combine the two objectives O1, O2 into a single loss function for minimisation, by using a regularization parameter $\boldsymbol{\mu}$.

$$E = (1-\mu) \sum_{i,j \in N_i} d(x_i, x_j) + \mu \sum_{i,j \in N_i, l, y_i \neq y_l} [d(x_i, x_j) + 1 - d(x_i, x_l)]_+$$

Large Margin Nearest Neighbour Metrics (LMNN)

- Now we make a few changes for making gradient computation easier.
- In t^{th} iteration, we can convert $D_t(x_i,x_j)$ as $\text{trace}(M_t C_{ij})$ where $C_{ij} = (x_i-x_j)(x_i-x_j)'$

$$
\begin{aligned}
d(x_i, x_j) &= (x_i - x_j)^T M (x_i - x_j) \\
&= trace((x_i - x_j)^T M (x_i - x_j)) \\
&= trace(M(x_i - x_j)^T (x_i - x_j)) \\
&= trace(MC_{ij})
\end{aligned}
$$

Large Margin Nearest Neighbour Metrics (LMNN)

- Now consider a set of triplets N_t . (i,j,k) is in N_t iff this set contributes to loss in O2.
- Then Gradient G_t can be obtained as follows.

$$G_t = \frac{\partial E}{M_t} = (1-\mu) \sum_{i,j \in N_i} C_{ij} + \mu \sum_{i,j,l \in N_i} [C_{ij} - C_{il}]$$

Large Margin Nearest Neighbour Metrics (LMNN)

Update Step:

The update simply subtracts the contributions from triples that are no longer active and adds the contributions of those that just became active from G_t and brings G_t+1:

$$G_{t+1} = G_t - \mu \sum_{i,j,l \in N_t - N_{t+1}} [C_{ij} - C_{il}] + \mu \sum_{i,j,l \in N_{t+1} - N_t} [C_{ij} - C_{il}]$$

Experimental Results

Observations

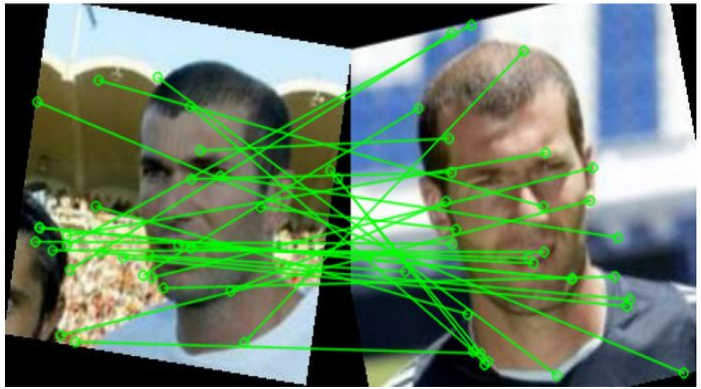
- In the paper, they claim that SIFT feature extraction performs well
  - We observe that ORB gives a more better accuracy compared to SIFT
- We performed face recognition using two following methods:
  - KNN Classification
  - LMNN + KNN
- We observe an accuracy of **33.3%** with KNN Classification, whereas an improved accuracy of **50%** with LMNN + KNN

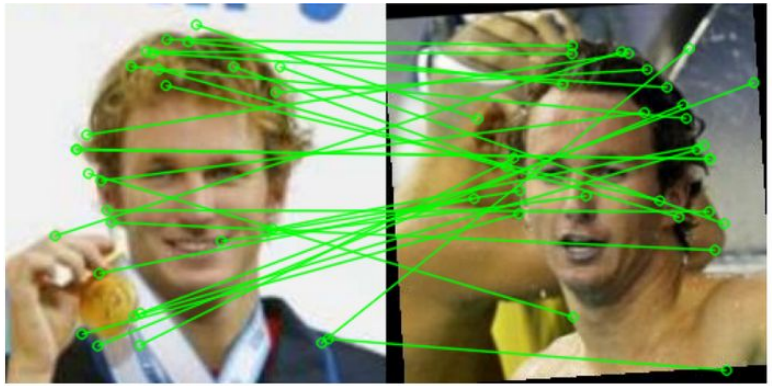
Experimental Results

Observations

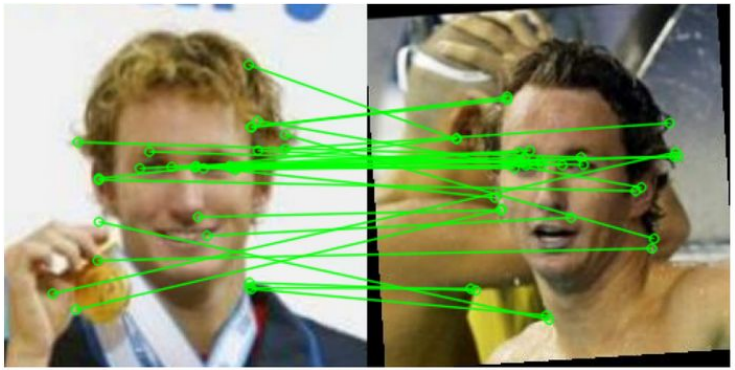
- We observed that when we perform KNN or KNN + LMNN directly without performing SIFT/ORB feature extraction, we get better results in terms of accuracies.
- We obtained an accuracy of **66.67%** with KNN Classification, and an improved accuracy of **73%** with LMNN + KNN .
- This shows that the SIFT/ORB step may not be working very well as expected.

SIFT Results Analysis





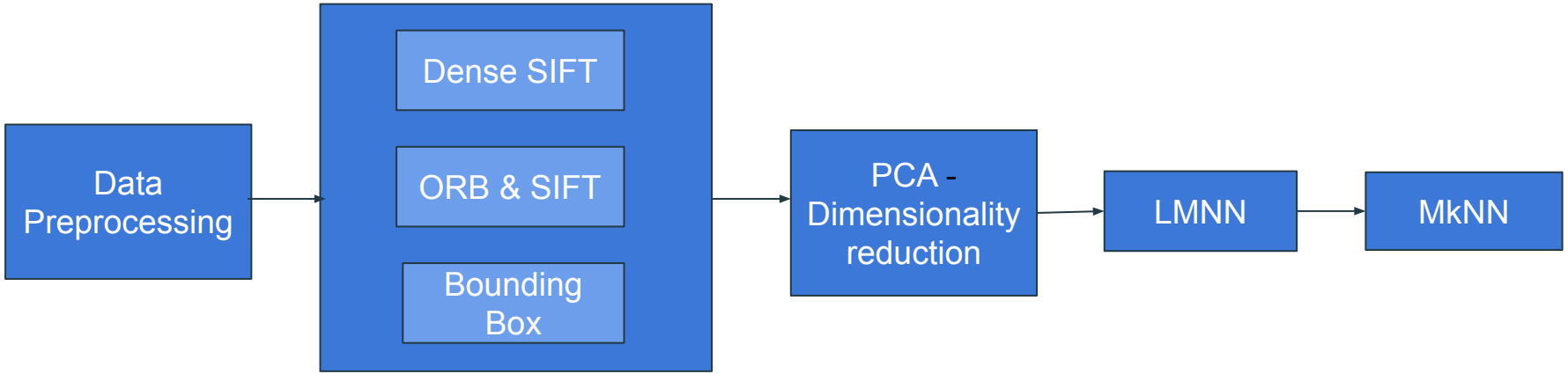
ORB Results Analysis



Interpretations

- As we see in the previous 4 images, our datasets of the same faces as well have noise in them. This is in terms of other objects/faces present in their background.
- Due to this reason, the SIFT/ORB descriptor is not able to focus only on extracting facial features that will aid out classification model.
- Small extensions can be done to this in terms of **providing a bounding box** to the images and then performing SIFT/ORB on that so that most of the descriptors returned are useful facial features.
- Due to variation in datasets and presence of noise, SIFT/ORB is not performing as well as expected.

Pipeline


Data Preprocessing
Dense SIFT
ORB & SIFT
Bounding Box
PCA - Dimensionality reduction
LMNN
MkNN

Feature Extraction - Improvised

Mid Evals - We have observed that normal sift/orb feature extractions deteriorated the Image Classification results.

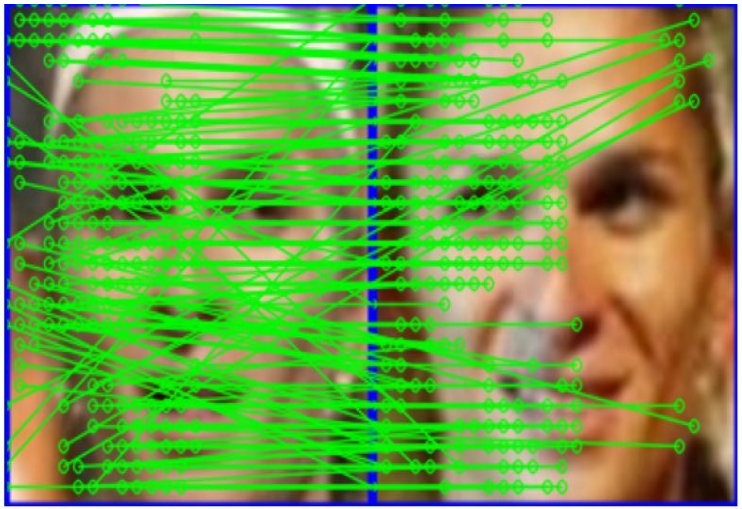
To Improve

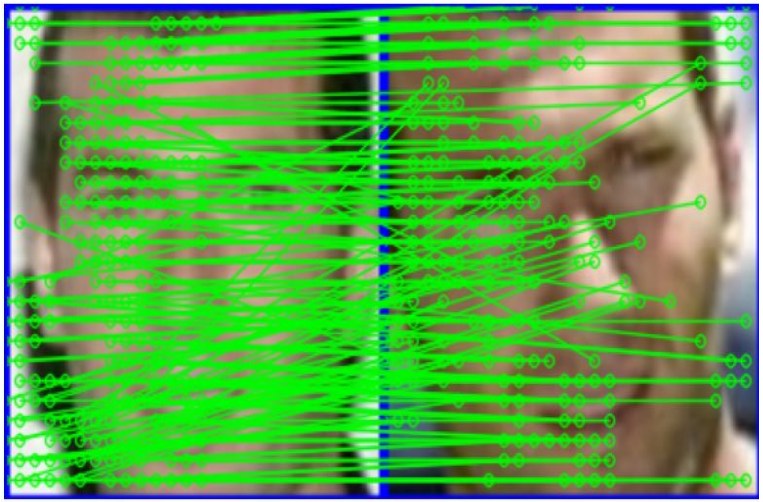
We modified the our traditional feature extraction methods as follows

- Using Dense sift feature extraction
- Bounding Box with dense sift
    - Here we used Viola-Jones algorithm to detect the face of the image and used dense sift on the cropped image to extract features

Dense-Sift with Bounding Box Results

We observed significantly improved results with the modified feature extraction methods





Observations

- As we observe that from the dense-sift using bounding box
  - The features extracted are more accurate
  - Descriptors are restricted to the face of the person
- When we use modified feature extraction for LMNN + KNN, we get improved results.
- The accuracy observed for KNN is 78.26% and for LMNN + KNN is 86.95%

Observed Accuracies

Feature	KNN	LMNN + KNN
Without Any Feature Extraction	66.67	73
Normal SIFT	33.3	50
Dense SIFT	68	80
Dese SIFT + Bounding Box	**78.26**	**86.95**

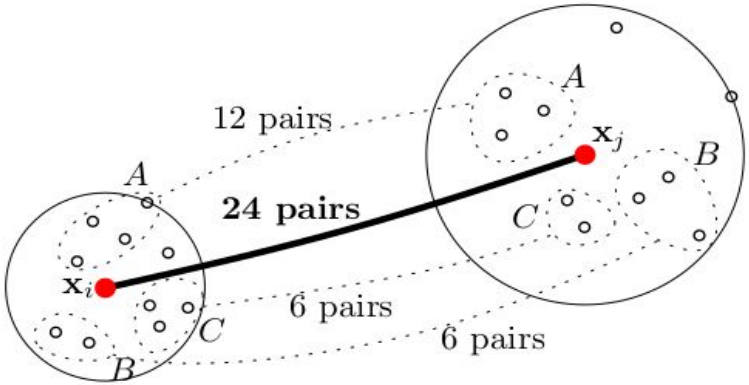
MKNN

- The probability of class C for x_i is $p(y_i=c|x_i)= n_c^i/k$ where n_c^i is the no.of neighbours of x_i of class c out of the k nearest neighbours of x_i.
- The marginal probability that we assign x_i and x_j to the same class using a kNN classifier is given by the below equation where n_c^i is the no.of neighbours of x_i of class c out of the k nearest neighbours of x_i and n_c^j is the no.of neighbours of x_i of class c out of the k nearest neighbours of x_j.Let the below equation be equation (1).

$$p(y_i = y_j|\mathbf{x}_i, \mathbf{x}_j) = \sum_c p(y_i = c|\mathbf{x}_i)p(y_j = c|\mathbf{x}_j)$$
$$= k^{-2}\sum_c n_c^i n_c^j.$$

MKNN

- The main goal is to find if the given pair of images belongs to the same class, regardless of which class that is and and even if the class is not represented in the training data.



Steps involved in MKNN

- For every point in the test data we find its k nearest neighbours in the training data.
- Now, for every point in the k nearest neighbors found for each point in the test data,we find its k nearest neighbours.
- Let A,B,C be the data points in the test data.
- Let k=3 and a1,a2,a3 are the 3 nearest neighbours of A.
- Now we find 3 nearest neighbors of a1,a2 and a3.

Steps involved in MKNN

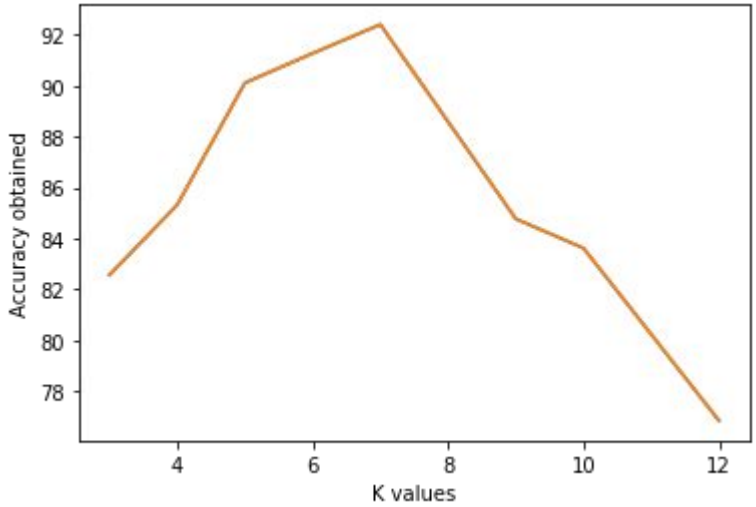
- Now using the above information and using the equation (1) given in the previous slide we calculate the marginal probability for the pairs {A,a1},{A,a2},{A,a3}.
- The maximum probability value is taken and then the label associated with that training data point is assigned to that particular test data point.
- In this way we for the other data points in the test data and find which class the point in the test data belongs to.

Observed Accuracies

Feature	KNN (using L2)	LMNN + KNN	MKNN (using L2)	LMNN + MKNN
Without Any Feature Extraction	66.67	73	68.12	75.25
Normal SIFT	33.3	50	43.75	57.125
Dense SIFT	68	80	70.65	83.25
Dense SIFT + Bounding Box	78.26	86.95	**85.45**	**91.95**

Varying k in MKNN

Accuracies for LMNN+MKNN vs k values



Project Deliverables

Mid Evaluation	Feature Extraction from Images using **SIFT - DONE**
	LMNN Implementation - DONE
End Evaluation	**MkNN** Implementation
	Improved Feature Extraction **(using dense-sift with bounding box over face)**
	Run different experiments to compare between L2, LMNN and different k values

THANK YOU