A Presentation
on
Person Re-Identification

By

Disha Shah (40011)

Umang Shah (40012)

Poras Vyas (40014)

M.Sc.(AI & ML) - 4

Under guidance of

Dr. Jyoti Pareek

(Department of Computer Science, Gujarat University)

Dr. Suchit Purohit

(Department of Computer Science, Gujarat University)

Department of Computer Science

Gujarat University

PERSON RE-IDENTIFICATION

INDEX

DEFINITION

- Person re-identification is the task of associating images of the same person taken from different cameras or from the same camera in different occasions.
- Humans can be easily identified by their face, height, build, clothing, hair style, walking pattern, etc.
- The subject is tracked across different places, either a posteriori or on-the-fly when they move through different locations.

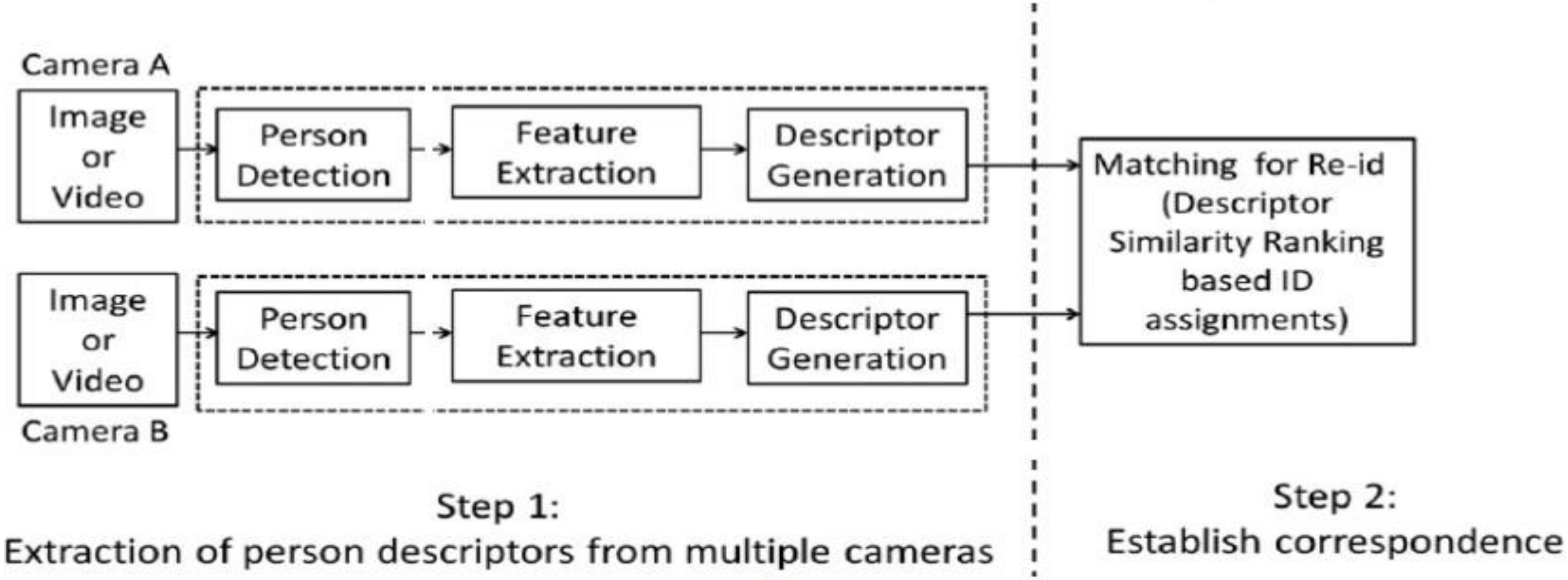
https://medium.com/@niruhan/a-practical-guide-to-person-re-identification-using-alignedreid-7683222da644

OBJECTIVE

- Person re-identification (Re-ID) aims at retrieving a person of interest across multiple non-overlapping cameras.
- In large distributed surveillance systems tracking an individual is a bit difficult task.
- The Person-ReID system helps to identify and track a person from different camera views by assigning a stable ID.
- The system is useful for uniquely identifying number of person on the campus.

https://medium.com/@niruhan/a-practical-guide-to-person-re-identification-using-alignedreid-7683222da644

PROCESS



https://medium.com/@niruhan/a-practical-guide-to-person-re-identification-using-alignedreid-7683222da644

APPLICATIONS

1. Security
2. Find lost person in public place.
3. Activity Tracking.

https://medium.com/@niruhan/a-practical-guide-to-person-re-identification-using-alignedreid-7683222da644

CHALLENGES

1. Detection of deformable object
2. Tracking multiple persons is also a challenging task.
3. Illumination changes.
4. Low resolution or inferior capturing devices.
5. Occlusion.
6. Uniform clothing

Indian Prime Minister Narendra Modi Practicing Yoga

https://medium.com/@niruhan/a-practical-guide-to-person-re-identification-using-alignedreid-7683222da644

LITERATURE REVIEW

Paper name : Person Re-identification: What Features Are Important?

Authors : Chunxiao Liu, Shaogang Gong, Chen Change Loy, and Xinggang Lin.

Approach : Color features are used to extract unique characteristics of a person.

Dataset : VIPeR and i-LIDS Multiple-Camera Tracking Scenario were used for evaluation. The VIPeR dataset contains 632 persons, each of which has two images captured in outdoor views.

https://www.researchgate.net/publication/236975411_Person_Re-identification_What_Features_Are_Important

LITERATURE REVIEW

Summary : In the technique discussed in this paper, clusters are created based on the color features extracted from an image and than from those features RankSVM method is used to select relevant features.

But color features may vary depending on there lighting conditions and camera angles.

Another problem may arise when two people with same clothes appear, as in that case, identifying one particular person among them will be difficult.

https://www.researchgate.net/publication/236975411_Person_Re-identification_What_Features_Are_Important

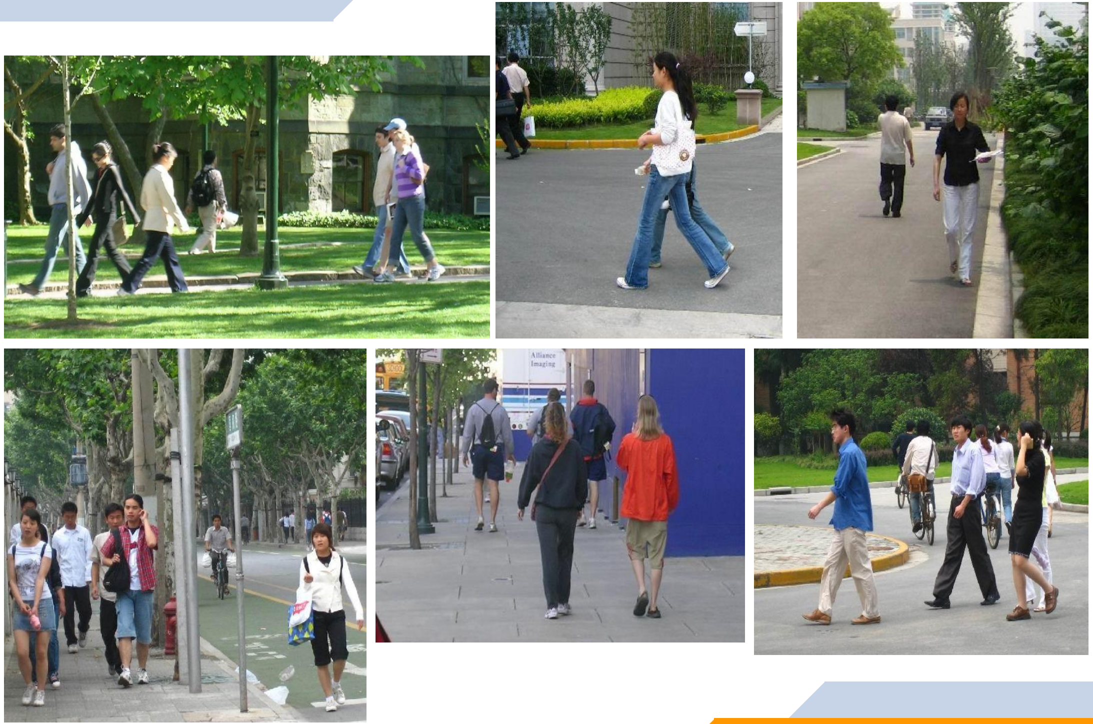
PERSON DETECTION

DATASET FOR DETECTION

- **Following datasets are used for person detection**
  1. Penn fudan
  2. Caltech pedestrian

PENN FUDAN

- Dataset was collected from University of Pennsylvania,Philadelphia, U.S. and Fudan University,Shanghai,China.
- Dataset format : images
- Images : 170
- Image size : 180x390
- Person : 345



https://www.cis.upenn.edu/~jshi/ped_html/

CALTECH PEDESTRIAN

- Dataset was collected from a vehicle driving through regular traffic in an urban environment.
- Dataset Format : Video
- Recording : 10 Hrs
- FPS : 30
- Frames : 250,000
- Frame Size : 640x480
- Person : 2300

http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/#:~:text=The%20Caltech%20Pedestrian%20Dataset%20consists,2300%20unique%20pedestrians%20were%20annotated

METHODS

Methods chosen to use for person detection.

1. HOG Person detector
2. Single shot detector.
3. Faster Region based CNN

HISTOGRAM OF ORIENTED GRADIENT

Person Detection

HISTOGRAM OF ORIENTED GRADIENT (HOG)

- HOG is a Feature Descriptor. That means it simplifies the image by extracting useful information and throwing away unwanted data.
- The HOG extracts features of an image by using gradient direction and gradient magnitude where gradient is a directional change in the intensity or colour in an image.

https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

HISTOGRAM OF ORIENTED GRADIENT (HOG)

Step 1 : Pre-processing

- Before loading image in hog function image is scaled to 64x128 for person detection.

Step 2 : Calculate the Gradient Vectors

- A gradient vector can be computed for every pixel in an image. It's simply a measure of the change in pixel values along the x-direction and the y-direction around each pixel.

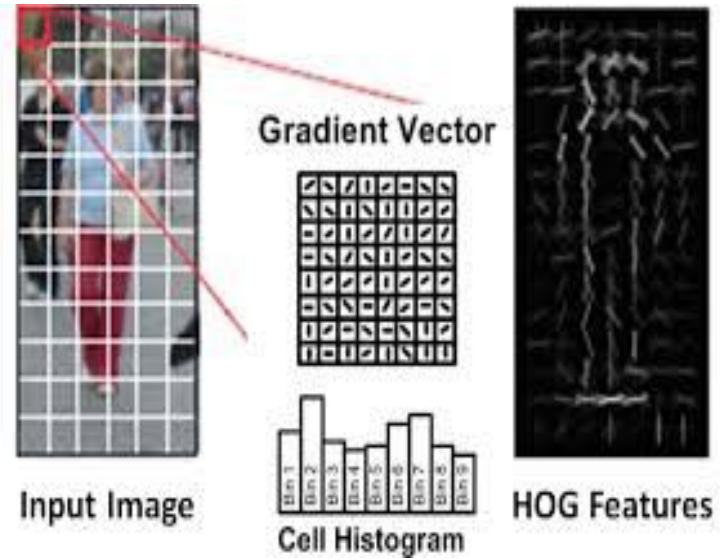
- Gradient is calculated by filtering the image using kernels.

https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

HISTOGRAM OF ORIENTED GRADIENT (HOG)

Step 3 : Calculate Histogram of Gradients in 16×16 cells

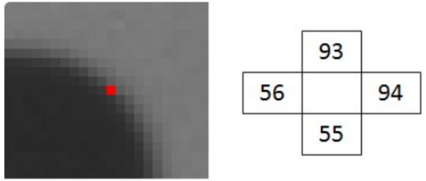
- The image is divided into 16×16 cells and a histogram of gradients is calculated for each 16×16 cells.
- The gradient of this image patch has 2 values (magnitude and direction) per pixel which adds up to 16 x 16 x 2 = 512 numbers.
- And this 512 numbers are converted into 9-bin histogram which can be stored as an array of 9 numbers.



https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

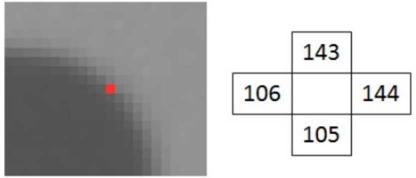
HISTOGRAM OF ORIENTED GRADIENT (HOG)

For better accuracy the HOG is contrast-**normalized**, this is done by calculating intensity over a larger area several cell known as block (16x16 Cell area), then this value is used to normalize all cells within that block. The normalized result gives better performance on variation in illumination and intensity.



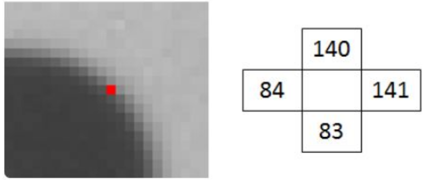
$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2 + (38)^2} = 53.74$$



$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2 + (38)^2} = 53.74$$



$$\nabla f = \begin{bmatrix} 57 \\ 57 \end{bmatrix}$$

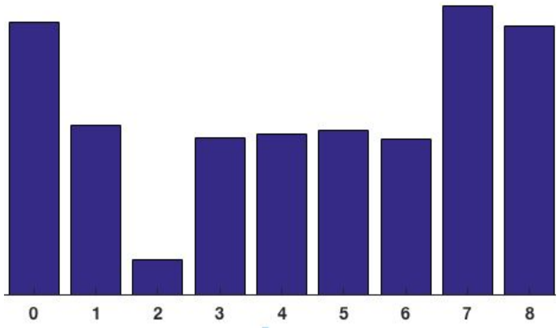
$$|\nabla f| = \sqrt{(57)^2 + (57)^2} = 80.61$$

https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

HISTOGRAM OF ORIENTED GRADIENT (HOG)

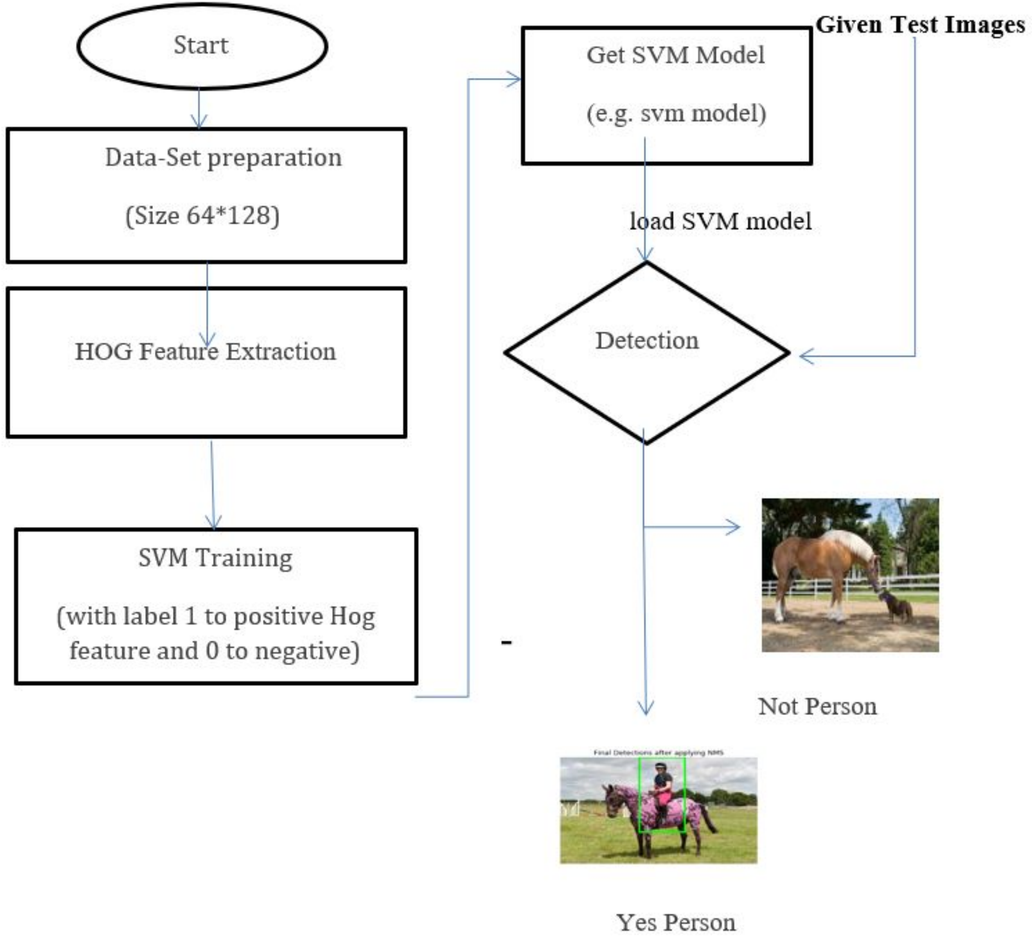
Step 3 : Calculate Histogram of Gradients in 16×16 cells

- This representation using histogram is more compact and more robust to noise. Individual gradients may have noise, but a histogram over 16×16 patch makes the representation much less sensitive to noise.
- The contributions of all the pixels in the 16×16 cells are added up to create the 9-bin histogram. For the patch above, it looks like this



https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

PROCEDURE


Start
Data-Set preparation
(Size 64*128)
HOG Feature Extraction
SVM Training
(with label 1 to positive Hog feature and 0 to negative)
Get SVM Model
(e.g. svm model)
Given Test Images
load SVM model
Detection
Not Person
Final Detections after applying NMS
Yes Person

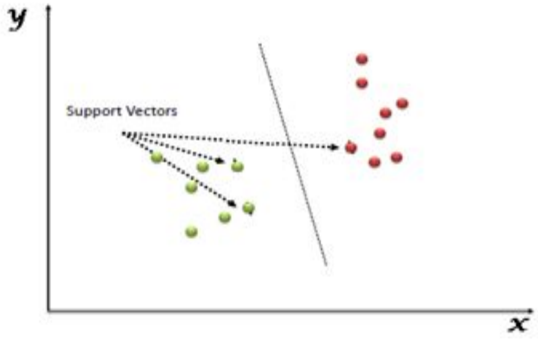
HOG CALCULATION

Detection window is moved from left to right and top to bottom. the window region is taken and an image classifier is applied to determine if the window has an object of interest – in this case, a person.

HISTOGRAM OF ORIENTED GRADIENT (HOG)

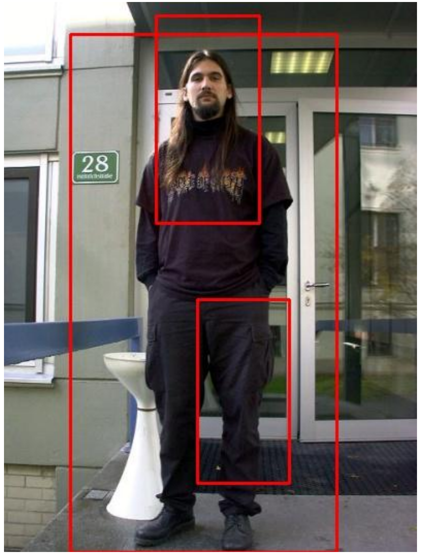
Support Vector Machine (SVM)

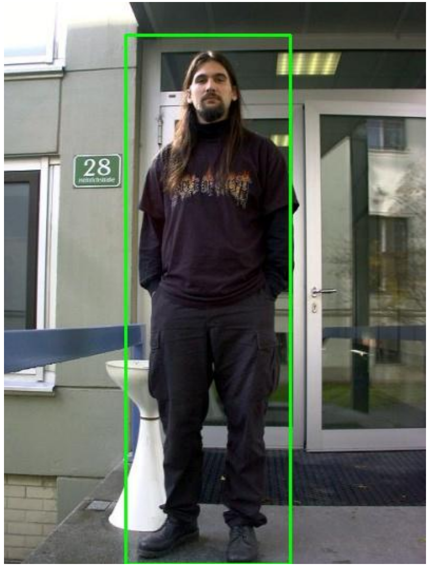
- In this algorithm, each data item is plotted as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Then, classification by finding the hyperplane is performed that differentiates the two classes very well
- In the linear classifier model, we assumed that training examples plotted in space. These data points are expected to be separated by an apparent gap. It predicts a straight hyperplane dividing 2 classes.



NON-MAXIMUM SUPPRESSION

- When using histogram of Oriented Gradients descriptor and a Linear Support Vector Machine for object classification, almost always multiple bounding boxes surrounding the object of interest are detected.
- applying non-maximum suppression helps ignore bounding boxes that overlap each other.





TESTING ENVIRONMENT

System Details :

- OS : Windows 8.1
- Processor : i7 5th gen
- Ram : 8 GB
- Image size : 400px width and height in that proportion

RESULTS OF HOG PERSON DETECTOR

Approx. time taken for prediction : 0.50 second

Occlusion

RESULTS OF HOG PERSON DETECTOR

Dark Lighting

RESULTS OF HOG PERSON DETECTOR

Scale Invariance

RESULTS OF HOG PERSON DETECTOR

Multiple Objects

TENSORFLOW OBJECT DETECTION STEPS

1. Train a CNN with regression(bounding box) and classification objective.
2. Normally their loss functions are more complex because it has to manage multiple objectives (classification, regression, check if there is an object or not)
3. Gather Activation from a particular layer (or layers) to infer classification and location with a Fully Connected layer or another CONV layer that works like a Fully Connected layer.
4. During prediction algorithms like non-maxima suppression is used to filter multiple boxes around same object.
5. During training time algorithms like IoU are used to relate the predictions during training the ground truth.

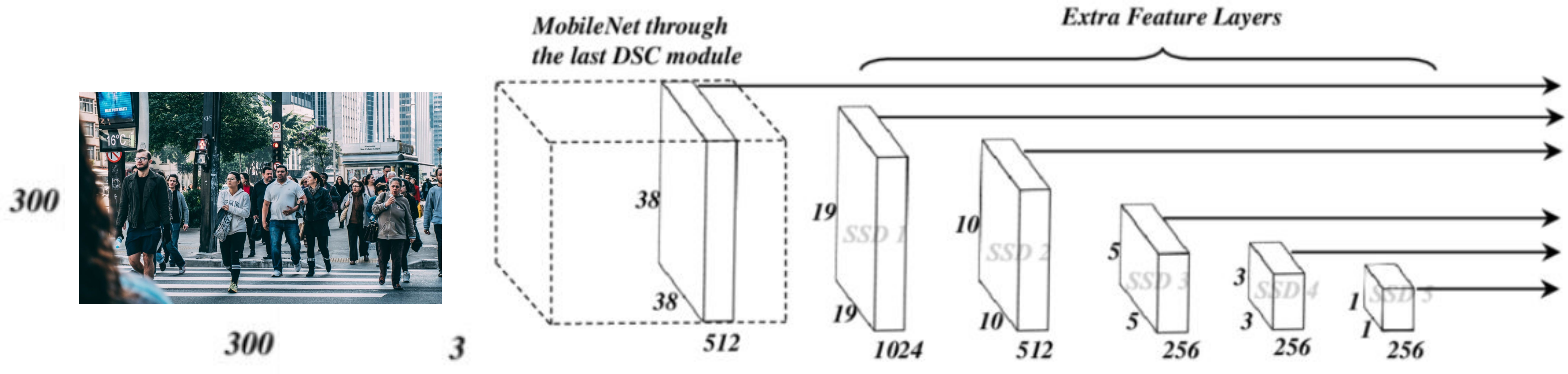
SINGLE SHOT DETECTOR

Person Detection

SINGLE SHOT DETECTOR

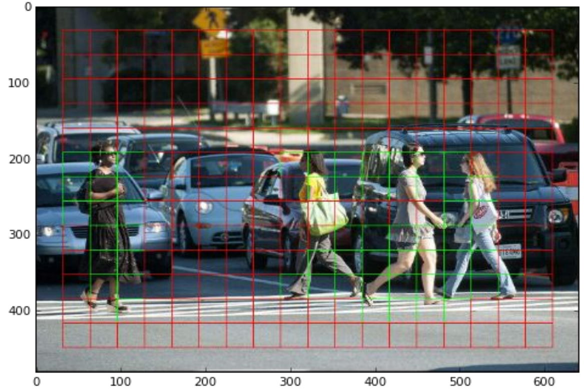
- Single Shot Detector was released in November 2016.
- As the name suggest it takes one single shot to detect multiple objects in an image as opposed to the region based architectures where the detection and classification occurs in two phases.

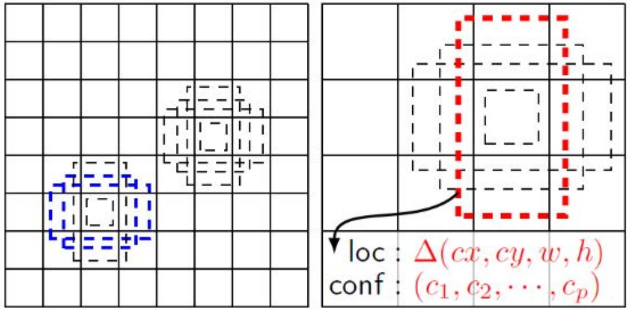
SINGLE SHOT DETECTOR



SINGLE SHOT DETECTOR

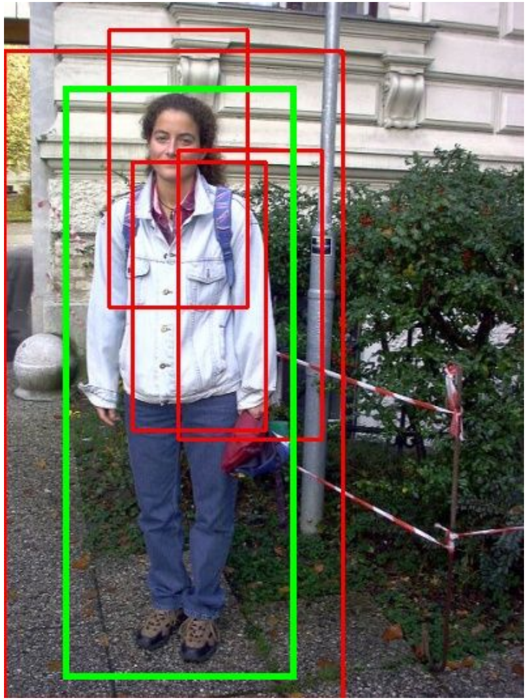
- After CNN layers, feature box of m x n x p is extracted from the image.
- For each location, k bounding boxes are found. These k bounding boxes have different sizes and aspect ratios.
- For each of the bounding box, c class scores are computed and 4 offsets relative to the original default bounding box shape.Thus, (c+4)kmn outputs are received.
- That’s why the algorithm is called “SSD: Single Shot MultiBox Detector”.

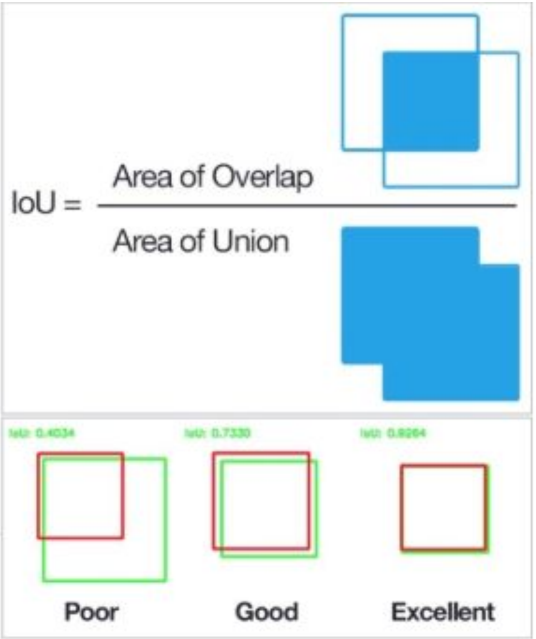




(b) 8×8 feature map (c) 4×4 feature map

WHAT IS IOU?




IoU =
Area of Overlap
Area of Union
IoU: 0.4034
IoU: 0.7330
IoU: 0.9264
Poor
Good
Excellent

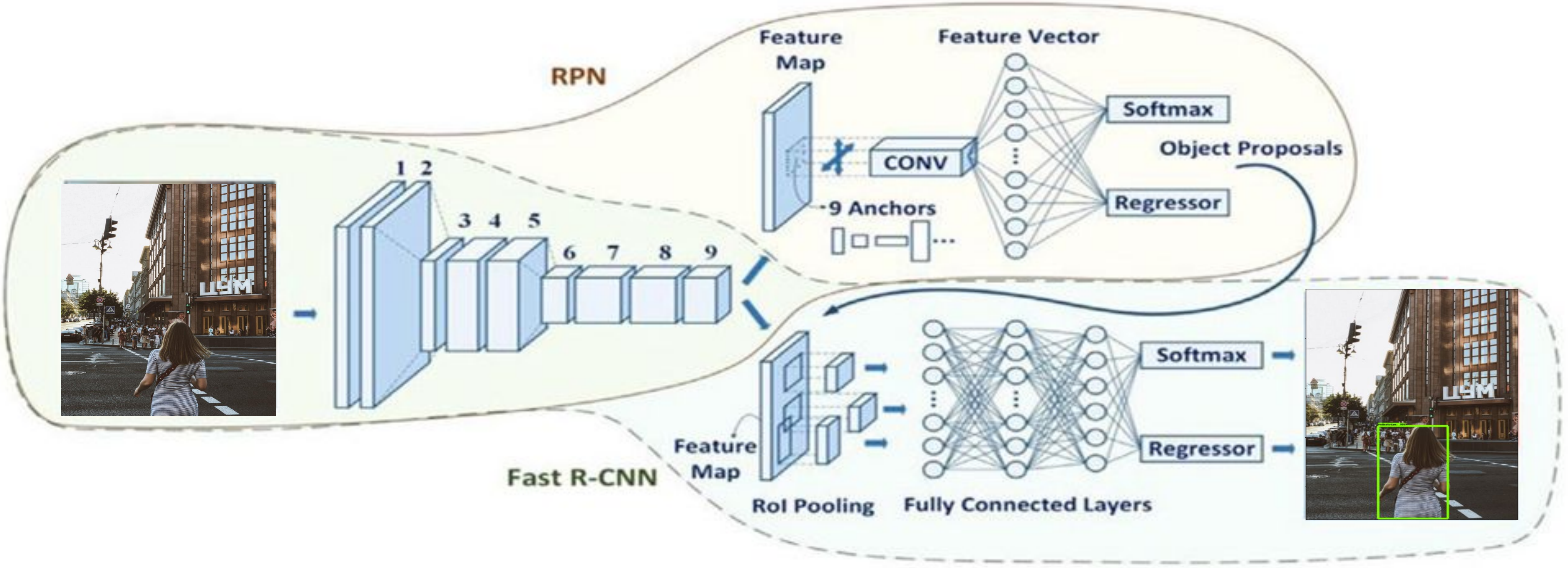
FASTER-RCNN

Person Detection

FASTER- RCNN

- Faster RCNN is named faster as such because it uses a Region Proposal Network (RPN) , unlike the selective search algorithm of its predecessors.
- Image or frame of a video is given as input and passed to the ConvNet which returns the feature map for that image.
- RPN is applied on these feature maps. This returns the object proposals along with their objectness score.

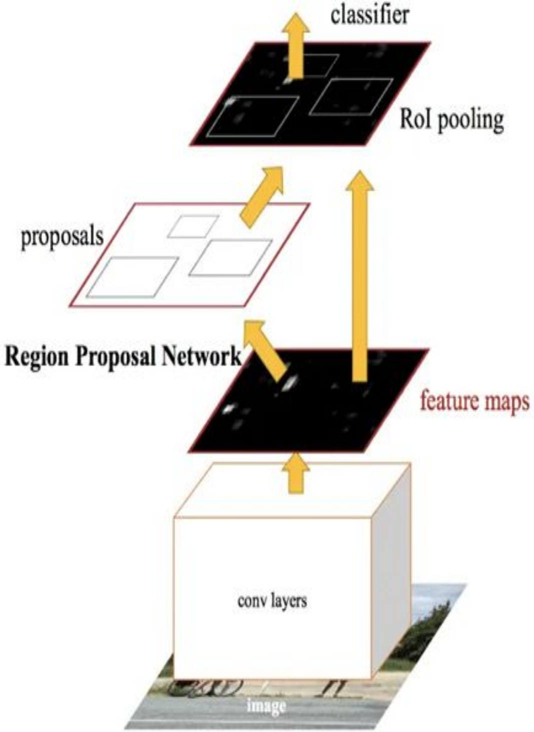
FASTER- RCNN



RPN
Feature Map
Feature Vector
CONV
9 Anchors
Softmax
Object Proposals
Regressor
1 2
3 4 5
6 7 8 9
Fast R-CNN
Feature Map
RoI Pooling
Fully Connected Layers
Softmax
Regressor

FASTER- RCNN

- A ROI pooling layer is applied on these proposals to bring down all the proposals to the same size.
- Finally, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.



WHAT IS LOSS?

- Loss is the penalty for a bad prediction. That is, loss is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples

LOSS IN OBJECT DETECTION

In Object detection, Loss is calculated in 2 parts:

1. **Localization loss**
   a. How close are the detected objects' bounding boxes to the ground truth boxes, here calculated by Huber loss(Smooth L1 loss)
   b. The smooth L1_loss is defined element-wise as :
   if |x| <= delta
   $0.5x^2$
   Else
   delta * (|x| - 0.5*delta)
   Where delta is some fixed value, or default 1.0, and x is the difference of actual and predicted answers.

LOSS IN OBJECT DETECTION

2. **Classification loss**
   a. A classification problem will have outputs in the form of how much percentage is the object likely to belong to a certain class. A perfect prediction is 1.0
   b. Therefore, classification loss is 1.0 - predicted probability for the target class.
   c. Tensorflow Object detection API uses Sigmoid Cross-entropy loss, calculated by :

      max(x, 0) - x * z + log(1 + exp(-abs(x))),

      where x is the logits value for each element and z is its label.

MEAN AREA PRECISION (mAP)

- Mean Area Precision (mAP) is a metric used to check the correctness of an Object detection model.
- **Precision** measures how accurate is your predictions. i.e. the percentage of your predictions are correct.
- **Recall** measures how good you find all the positives

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

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

TP = True positive

TN = True negative

FP = False positive

FN = False negative

MEAN AREA PRECISION (mAP)

- If we plot a graph with x-axis having recall and y-axis having precision, the general definition for the **Average Precision**(AP) is finding the area under the precision-recall curve.

- Mean of all APs of the examples under consideration is considered as the **Mean Average Precision (mAP).**

RESULTS

SSD and Faster RCNN

PennFudan Dataset

	SSD	FRCNN
Occlusion		
Lighting		

PennFudan Dataset

	SSD	FRCNN
Scale Variance	 	 
Multiple Subjects		

RESULTS

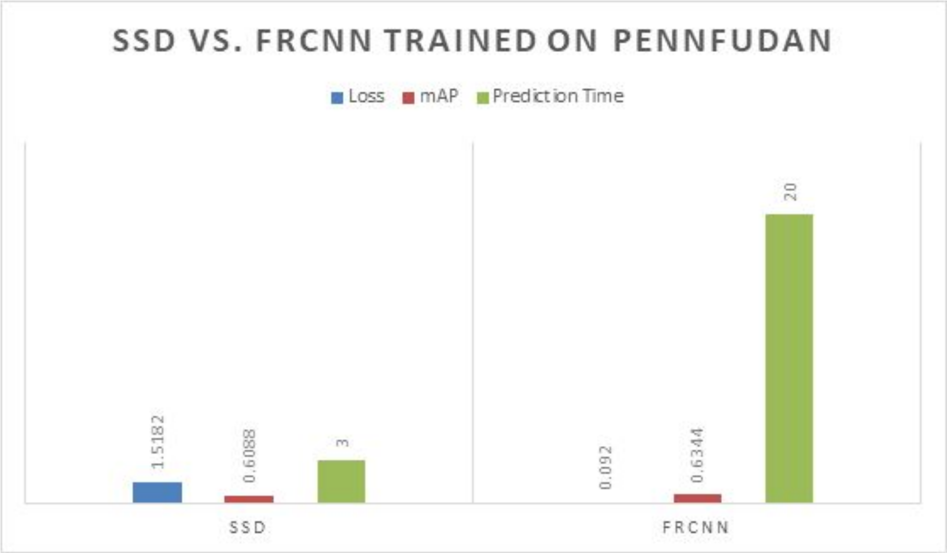
Caltech Pedestrian Dataset

	SSD	FRCNN
Occlusion		
Lighting		

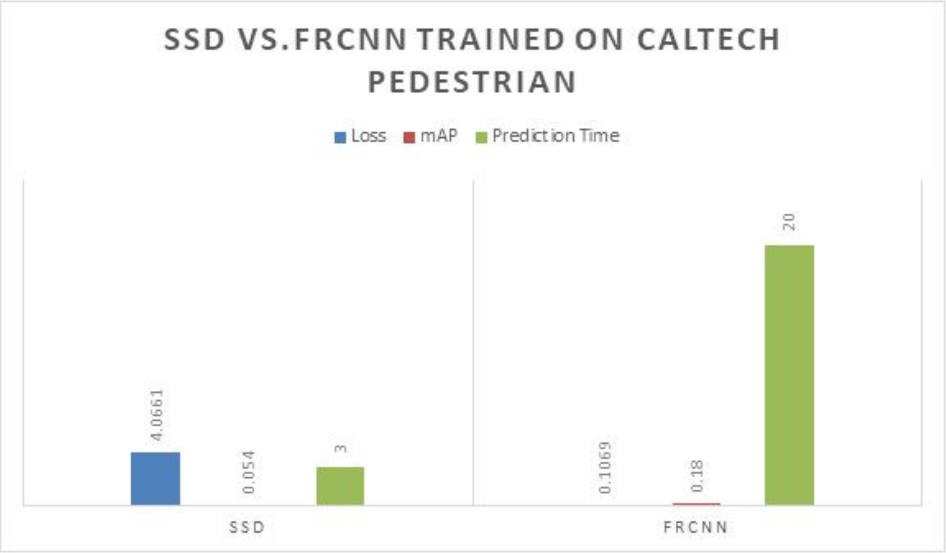
Caltech Pedestrian Dataset

	SSD	FRCNN
Scale Variance	 	 
Multiple Subjects		

RESULTS



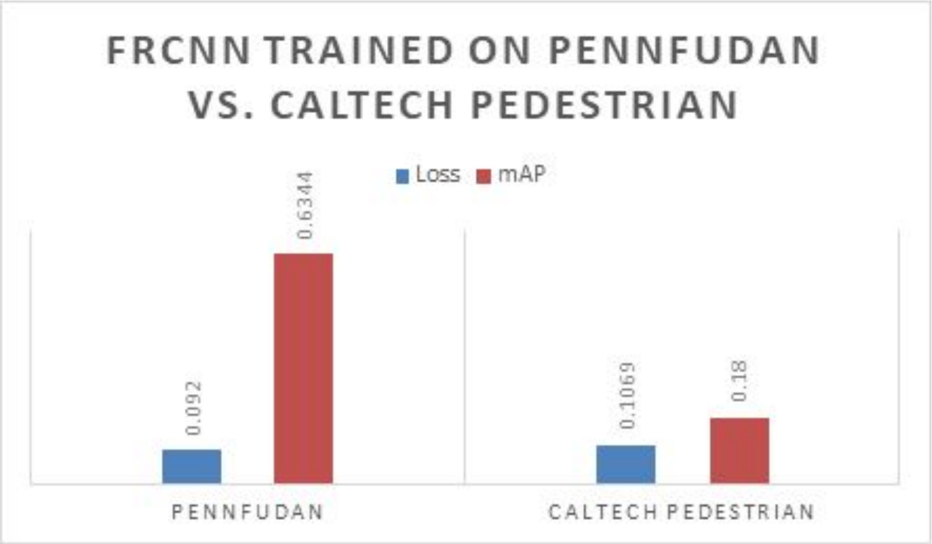
SSD VS. FRCNN TRAINED ON PENNFUDAN
Loss
mAP
Prediction Time
1.5182
0.6088
3
0.092
0.6344
20
SSD
FRCNN



SSD VS.FRCNN TRAINED ON CALTECH PEDESTRIAN
Loss
mAP
Prediction Time
4.0661
0.054
3
0.1069
0.18
20
SSD
FRCNN

CONCLUSION ON DETECTION ALGORITHM

- SSD algorithm is fast for person detection but it is not able to identify the person in all ideal condition.
- Whereas FRCNN takes much time for detection but it is able to identify the person in all ideal condition.



COLAB FILE LINK

Hog Detector

Tensorflow Object detection

Feature Extraction

FEATURE EXTRACTION

Feature Extraction methods:

1. Color based feature extraction
2. Shape based feature extraction
3. Convolution neural network

COLOR BASED FEATURE EXTRACTION

COLOR BASED FEATURE EXTRACTION

Two methods to compare histogram :

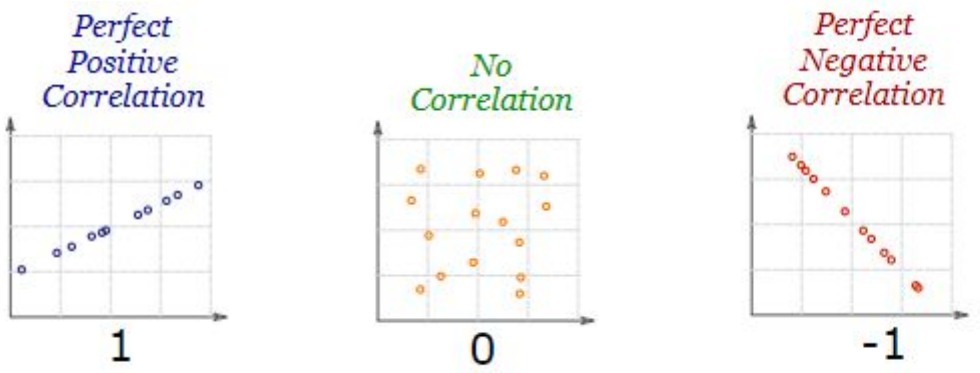
1. Opencv histogram comparison method
   a. Correlation
   b. Chi-squared
   c. Intersection
   d. Bhattacharya
2. SciPy distance metrics
   a. Euclidean
   b. manhattan

COLOR BASED FEATURE EXTRACTION

Opencv histogram comparison method - Correlation

Correlation can have one of these values:

- **1** is perfect positive correlation.
- **0** is no correlation
- **-1** perfect negative correlation.



COLOR BASED FEATURE EXTRACTION

Opencv histogram comparison method - Correlation

- Correlation (method=CV_COMP_CORREL)

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}}$$

- Where, $\bar{H_k} = \frac{1}{N} \sum_J H_k(J)$

- And N is a total number of histogram bins.

The Formula for Correlation Is

$$r = \frac{\sum (X - \overline{X})(Y - \overline{Y})}{\sqrt{\sum (X - \overline{X})^2} \sqrt{(Y - \overline{Y})^2}}$$

where:

r = the correlation coefficient

$\overline{X}$ = the average of observations of variable X

$\overline{Y}$ = the average of observations of variable Y

COLOR BASED FEATURE EXTRACTION

Query Images

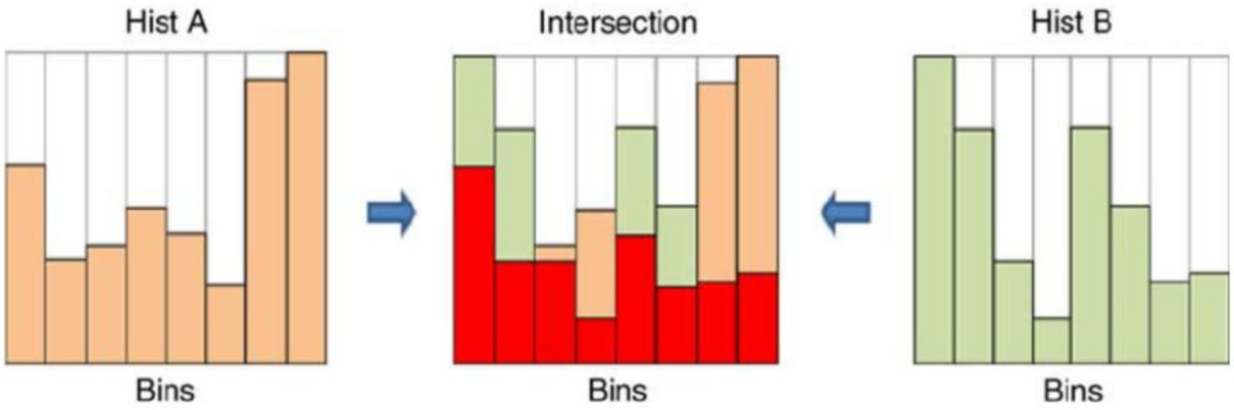
Correlation

Value near 1 indicates more similarity.

Opencv histogram comparison method - Intersection

- Histogram intersection calculates the similarity between two histograms.
- Intersection (method=CV_COMP_INTERSECT)
- $$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

COLOR BASED FEATURE EXTRACTION



Histogram Intersection Similarity Method (HISM). Histograms are normalized so that their minimum value is 0 and their maximum value is 1. Histograms A (left) and B (right) are examples of target and local feature histograms, respectively. Their intersection (center) consists of the smaller of the two values in corresponding bins. The size of the red area is our measure of similarity between target and local histograms.

COLOR BASED FEATURE EXTRACTION

Intersection

Query Images

Higher value indicates more similarity.

Opencv histogram comparison method - Chi-square

- Chi square is a method used in statistics that calculates the difference between observed and expected data values. It is used to determine how closely actual data fit expected data.
- Chi-Square (method=CV_COMP_CHISQR)
- $$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

COLOR BASED FEATURE EXTRACTION

Chi-Squared

Query Images

small value indicates more similarity.

Opencv histogram comparison method - Hellinger

- Hellinger distance is measure of similarity between two histograms.
- Bhattacharyya distance (method=CV_COMP_BHATTACHARYYA or method=CV_COMP_HELLINGER). In fact, OpenCV computes Hellinger distance, which is related to Bhattacharyya coefficient.
- $$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H_1}\bar{H_2}N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

COLOR BASED FEATURE EXTRACTION

Hellinger

Query Images

small value indicates more similarity.

SciPy distance metrics - Euclidean Distance

- Computes euclidean distance between two histograms.
- Calculates difference between a point in two histogram by following formula
- Scipy.spatial (distance.euclidean)
- $$\sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

COLOR BASED FEATURE EXTRACTION

Euclidean

Query Images

Smaller distance shows more similarity

SciPy distance metrics - Manhattan Distance

- Computes manhattan distance between two histograms.
- Scipy.spatial (distance.cityblock)
- $d(H_1, H_2) = \sum_{l} (H_1(\mathrm{l}) - H_2(\mathrm{l}))$

COLOR BASED FEATURE EXTRACTION

Manhattan

Query Images

Smaller distance shows more similarity

SHAPE BASED FEATURE EXTRACTION

SHAPE BASED FEATURE EXTRACTION

- The SURF method (Speeded Up Robust Features) is a fast and robust algorithm for local, similarity invariant representation and comparison of images.
- The main interest of the SURF approach lies in its fast computation of operators using box filters, thus enabling real-time applications such as tracking and object recognition.

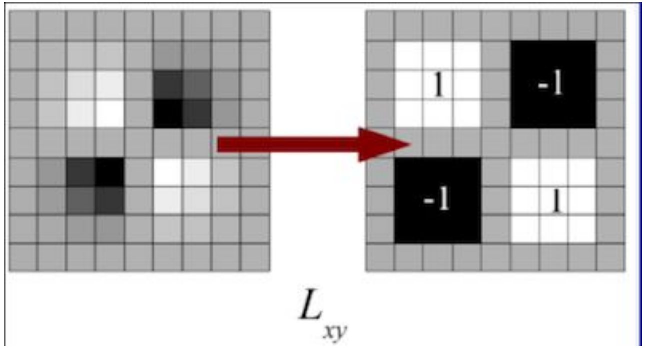
SHAPE BASED FEATURE EXTRACTION

- Surf uses the **Hessian matrix** because of its good performance in computation time and accuracy. Rather than using a different measure for selecting the location and the scale , surf relies on the determinant of the Hessian matrix for both. Given a pixel, the Hessian of this pixel is something like:

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

SHAPE BASED FEATURE EXTRACTION

- To calculate the determinant of the Hessian matrix, first apply convolution with Gaussian kernel, then second-order derivative.



- We denote these approximations by Dxx, Dyy, and Dxy. Now we can represent the determinant of the Hessian (approximated) as:

$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2.$$

SHAPE BASED FEATURE EXTRACTION

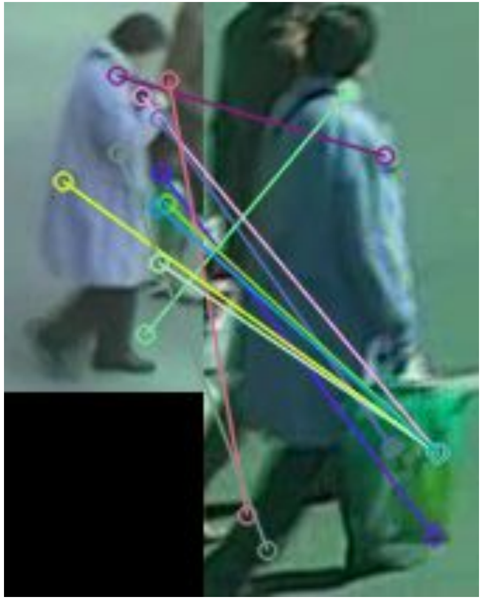
- For SURF feature extraction, we have used OpenCV library functions,where we give the size of hessian filter,which acts as a controlling factor for no. of keypoints.

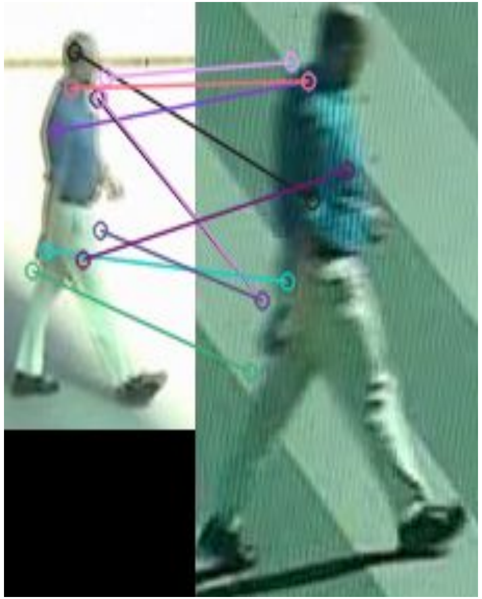
```
def feature_extract(image_obj):
  surf_obj = cv2.xfeatures2d.SURF_create(100)
  kp,ds = surf_obj.detectAndCompute(image_obj,None)
  return [kp,ds]
```

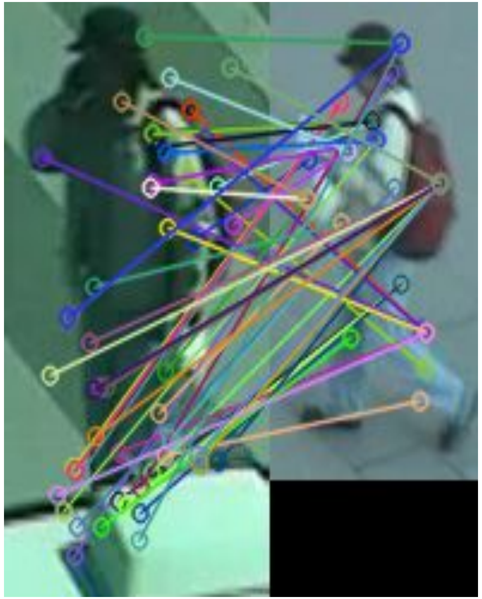
SHAPE BASED FEATURE EXTRACTION

- The extracted features are then compared using Brute Force Matcher given in OpenCV library.
- It uses KNN to find k no. of vectors by taking query and train descriptors as input and finding matches.
- Each match found using this BFMatcher obj. has an attribute ‘distance’.
- The matches from the k vectors having distance less than certain threshold can be considered as best
- The distance here is by default calculated using L2Norm(Euclidean), but has an option to use L1Norm(Manhattan) too.

SHAPE BASED FEATURE EXTRACTION







COLAB FILE LINK

Color based feature extraction

Shape based feature extraction

CONVOLUTIONAL NEURAL NETWORK

Transfer Learning

- Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.
- Transfer Learning differs from traditional Machine Learning in that it is the use of pre-trained models that have been used for another task to jump start the development process on a new task or problem.

CONVOLUTIONAL NEURAL NETWORK

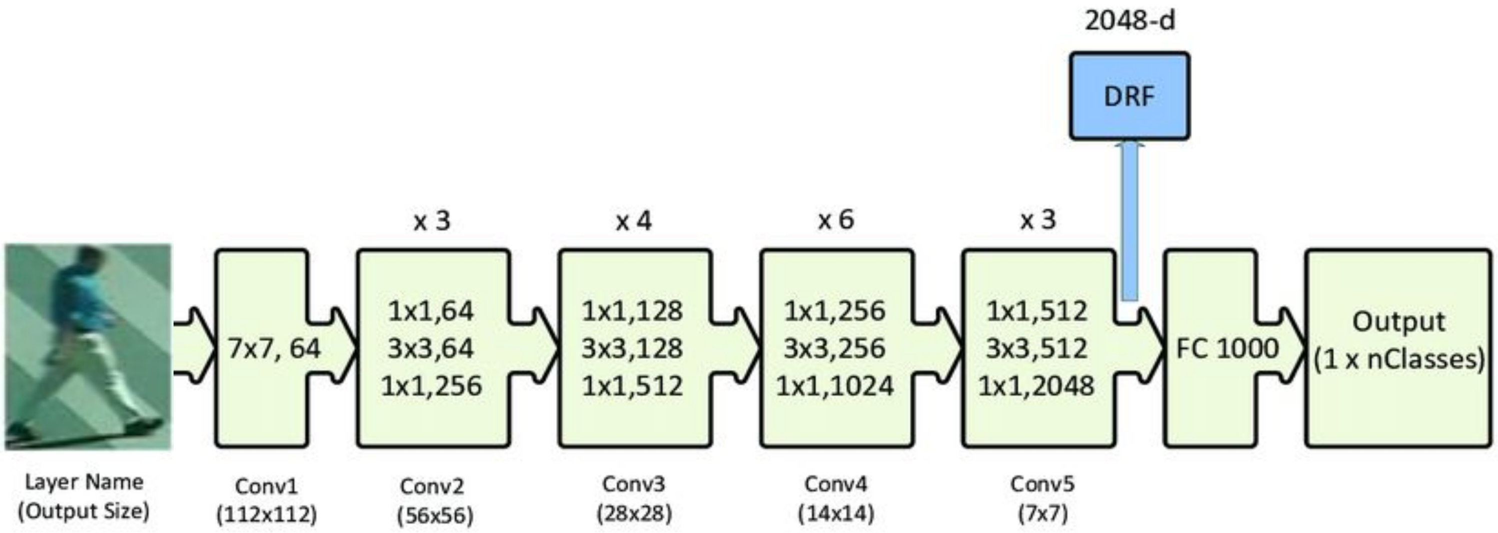
- **There are two ways to do transfer learning:**
  1. Feature Extraction from pre-trained model and then training a classifier or applying distance-based measure on top of it.
  2. Fine tuning the pre-trained model keeping learnt weights as initial parameters.

CUHK DATASET

- This dataset is collected from the Chinese University of Hong Kong (CUHK) campus.
- The dataset contains 843 identities having total 8177 images.
- The dataset was collected using 2 cameras.

https://www.ee.cuhk.edu.hk/~xgwang/CUHK_identification.html

CONVOLUTIONAL NEURAL NETWORK



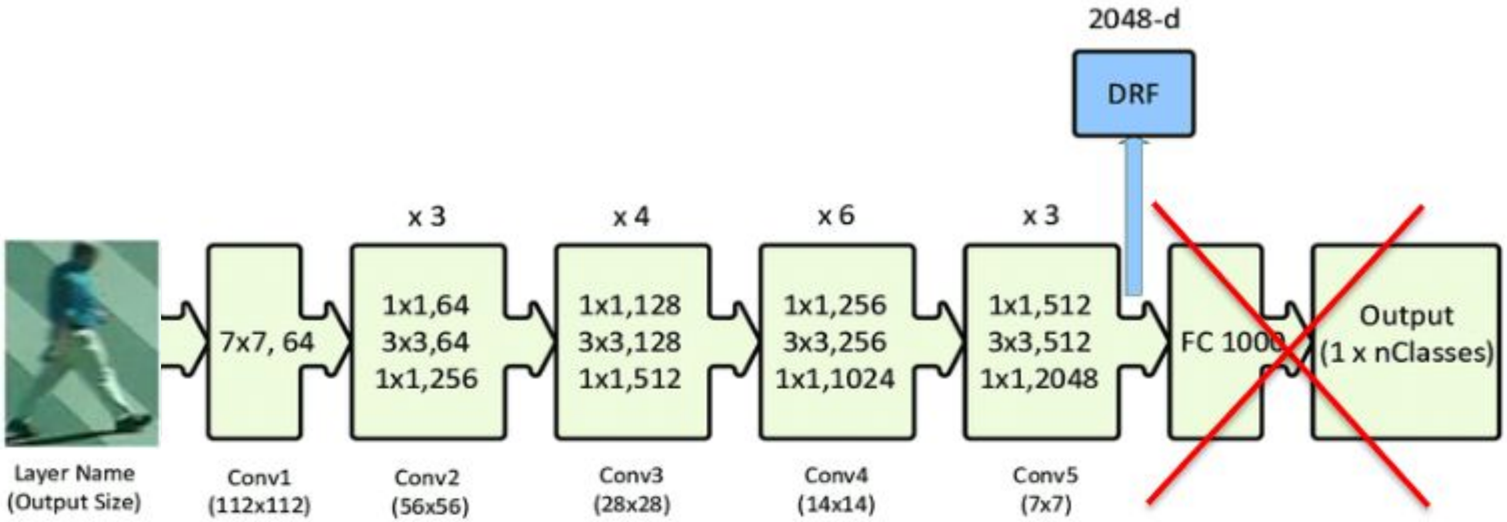
Resnet50

CONVOLUTIONAL NEURAL NETWORK

- We have used Resnet50 architecture pre-trained on image-net dataset as initial parameters.
- Last convolution layer of Resnet50 had dimensionality of (7,7,2048).
- We added Global Average Pooling layer for averaging all the filters so that we can get the 2048 dimensional vector and at the end we added classification layer which classifies the identities.

Fan, Hehe, et al. "Unsupervised person re-identification: Clustering and fine-tuning." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.4 (2018): 1-18.

CONVOLUTIONAL NEURAL NETWORK



Resnet50

CONVOLUTIONAL NEURAL NETWORK

- For testing we removed the classification layer and used the global average pooling layer which gives 2048 features for each person.
- By doing this we can use this trained network on any person.

CONVOLUTIONAL NEURAL NETWORK

- We divided the test images into query images and registered images so that query images' features can be extracted and compared with registered images' features.
- Euclidean distance is calculated between each query from the query feature list with all the registered feature list. The feature vector with least distance was used for identifying the query.
- This is how we got predicted data. The predicted data and the true data was used for calculating accuracy.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

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

CONVOLUTIONAL NEURAL NETWORK

- Feature vector thus extracted many contain values in the range (-infinity to +infinity)
- To control its output, we normalize the feature vector so it can work in different lighting and color conditions.

$$U = \frac{V}{||V||} \quad where \quad ||V|| = \sqrt{X^2 + Y^2}$$

CONVOLUTIONAL NEURAL NETWORK

	Training	Testing
Accuracy	99.71	97.41
Identities	743	100

CONVOLUTIONAL NEURAL NETWORK

DISTANCES

Query Images

CONVOLUTIONAL NEURAL NETWORK

DISTANCES

Query Image

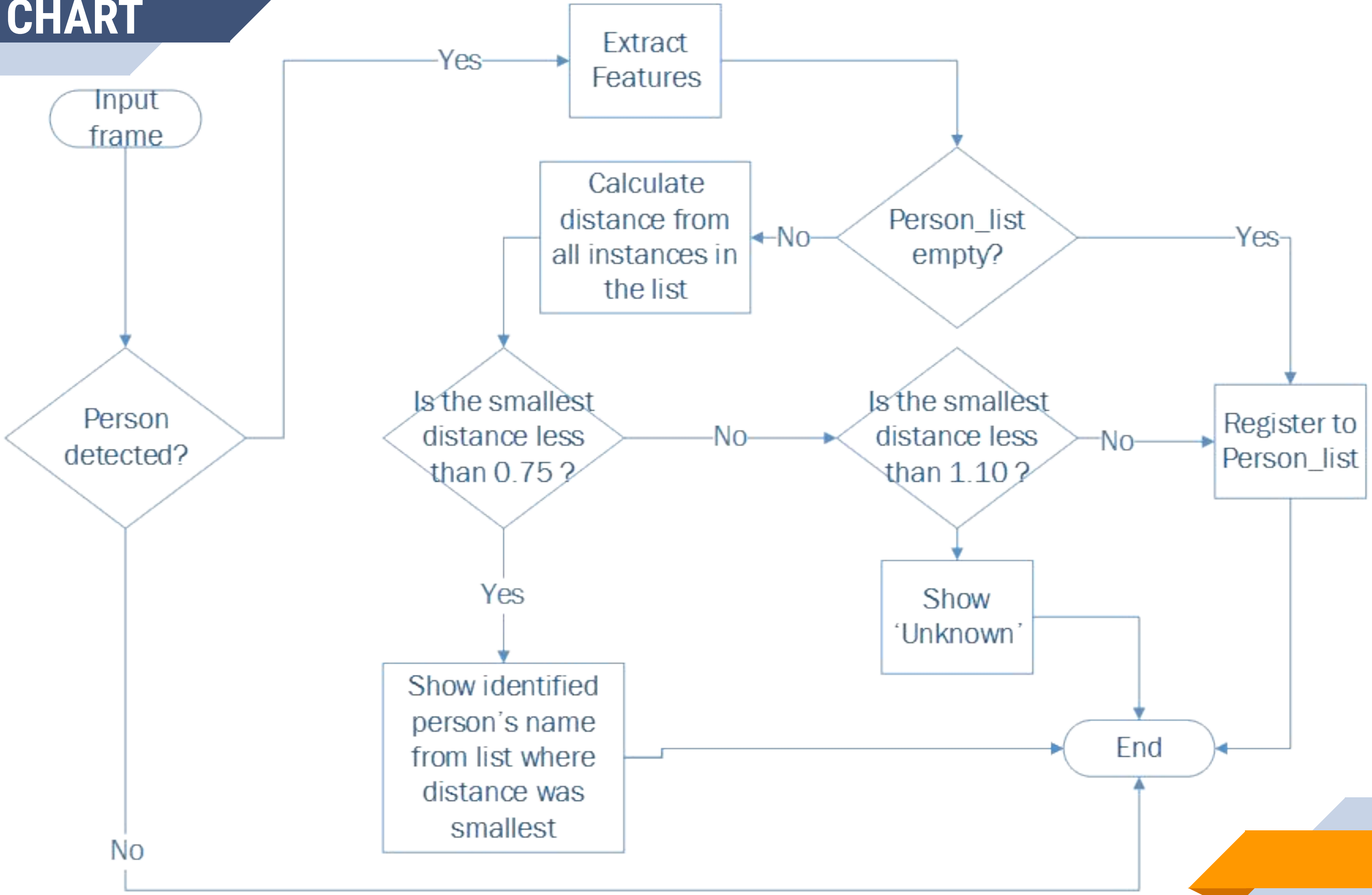
CONVOLUTIONAL NEURAL NETWORK

DISTANCES

Query Images

PERSON RE-IDENTIFICATION SYSTEM

FLOW CHART



Input frame
Person detected?
Yes
No
Extract Features
Person_list empty?
No
Yes
Calculate distance from all instances in the list
Is the smallest distance less than 0.75 ?
No
Yes
Is the smallest distance less than 1.10 ?
No
Register to Person_list
Show 'Unknown'
Show identified person's name from list where distance was smallest
End

PERSON RE-IDENTIFICATION SYSTEM

- This system first registers a person if the ‘person_list’ is empty.
- Then on every occurrence of the person object it calculates the feature of detected person.
- Adds the new person if the distance of that person with every person from the list is larger than 1.10
- Identifies a person if the lowest distance of the current feature vector with other feature vectors from the list is less than 0.75

PERSON RE-IDENTIFICATION SYSTEM

This system is tested on a device having following configurations

Processor	Intel i7 5th Generation
RAM	8 GB
GPU	NVIDIA GeForce GTX 960M

Sources for results:

- There were in total 5 videos taken, using 2 different devices
  1. iPhone Xs, Camera: 12 Megapixel
  2. Samsung Galaxy S8. Camera: 12 Megapixel

RESULTS

Person List (Empty)

Register

Person List (person 0) (person 1)

Identify

RESULTS

Clothing Change

Register

Person List
(person 0)
(person 1)
(person 2)

Identify
as new

New clothing, New person

RESULTS

Background Change

Identify

Person List
(person 0)
(person 1)
(person 2)

Identify

Occlusion

RESULTS

Different angles

Identify

Person List
(person 0)
(person 1)
(person 2)

Identify

Different angles

RESULTS

Different lighting

Not Identified

Person List
(person 0)
(person 1)
(person 2)

Not Identified

Different lighting

PERSON RE-IDENTIFICATION SYSTEM

- If we want to identify the person with different postures and in various lighting conditions, then it is preferable to take more pictures of the same person in different angles.
- By doing this each person will have higher possibilities of being recognized in various ideal conditions

PERSON RE-IDENTIFICATION SYSTEM

Images with different angles and lighting conditions for better results.

RESULTS

Different lighting

Identify

Person List
(person 0)
(person 1)
(person 2)

Identify

Different lighting

Problems with the system:

1. The system identifies the person based on the color of their clothes. When clothes are different the system is unable to identify the person as same person.
2. For best re-identification, different examples of the same person need to be taken or the data of the same person has to be augmented, in order to avoid lack of information.

CONCLUSION

- Person Re-Identification system works well when a person does not change clothes.
- So, this system will work well at places like shops, malls, offices etc. as a person will be wearing same clothes.
- The system is useful for counting person in a room keeping track of their movements in a place etc.
- This system will face problem in security surveillance, if person changes clothes then identifying him/her will be difficult. So, for such conditions system needs to be modified accordingly.

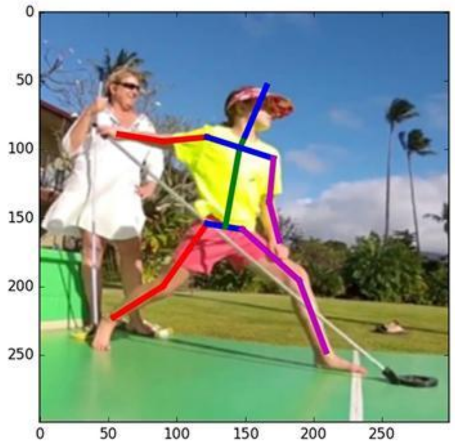
GAIT RECOGNITION (FUTURE WORK)

- Gait Recognition is human motion estimator.
- Walking pattern of a person is captured for recognition.
- This type of biometric approach is considered non-invasive, since it is performed at a distance, and does not require the cooperation of the subject that has to be identified.

Castro, F.M., Marín-Jiménez, M.J., Mata, N.G. and Muñoz-Salinas, R., 2017. Fisher motion descriptor for multiview gait recognition. International Journal of Pattern Recognition and Artificial Intelligence, 31(01), p.1756002.

GAIT RECOGNITION (FUTURE WORK)

- **Steps of gait recognition:**
  1. Body parts of human are detected to track their motion
  2. Gait cycle is given as input to generate features. These features are stored in the database.
  3. At the time of recognition stored features and current features are compared to identify a person.



Castro, F.M., Marín-Jiménez, M.J., Mata, N.G. and Muñoz-Salinas, R., 2017. Fisher motion descriptor for multiview gait recognition. International Journal of Pattern Recognition and Artificial Intelligence, 31(01), p.1756002.

GAIT RECOGNITION (FUTURE WORK)

- **Advantage of Gait Recognition:**
  1. This system is not dependent on clothing colors, same person with different clothes can be identified.
  2. Just by having video of a person walking, we can identify him/her in future videos.
- **Disadvantage of Gait Recognition:**
  1. This system has a limitation that if a person changes his/her walking pattern intentionally, system will fail to recognize that person.

Castro, F.M., Marín-Jiménez, M.J., Mata, N.G. and Muñoz-Salinas, R., 2017. Fisher motion descriptor for multiview gait recognition. International Journal of Pattern Recognition and Artificial Intelligence, 31(01), p.1756002.

REFERENCES

- **Research Paper**

  Liu, Chunxiao, et al. "Person re-identification: What features are important?." *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2012.

- **Color based feature extraction**

  https://link.springer.com/content/pdf/10.1007%2F978-3-319-10590-1_35.pdf

- **Shape based feature extraction**

  https://www.researchgate.net/publication/241166590_Object_Reidentification_in_Multiple_Cameras_System

  https://docs.opencv.org/3.4/d5/df7/classcv_1_1xfeatures2d_1_1SURF.html

- **Histograms of Oriented Gradients**

  https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

REFERENCES

- **Penn-Fudan Dataset**

  https://www.cis.upenn.edu/~jshi/ped_html/

- **Caltech Pedestrian Dataset**

  http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/#:~:text=The%20Caltech%20Pedestrian%20Dataset%20consists,2300%20unique%20pedestrians%20were%20annotated

- **Person re-identification**

  https://medium.com/@niruhan/a-practical-guide-to-person-re-identification-using-alignedreid-7683222da644

  Fan, Hehe, et al. "Unsupervised person re-identification: Clustering and fine-tuning." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.4 (2018): 1-18.

REFERENCES

- **Transfer Learning**

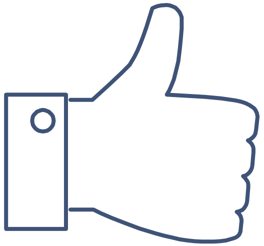
  https://medium.com/kansas-city-machine-learning-artificial-intelligen/an-introduction-to-transfer-learning-in-machine-learning-7efd104b6026

- **CUHK Dataset**

  https://www.ee.cuhk.edu.hk/~xgwang/CUHK_identification.html

- **Gait Recognition**

  Castro, F.M., Marín-Jiménez, M.J., Mata, N.G. and Muñoz-Salinas, R., 2017. Fisher motion descriptor for multiview gait recognition. International Journal of Pattern Recognition and Artificial Intelligence, 31(01), p.1756002.



THANKS!

Any questions?