

Title: Helmet Detection of Motor Vehicles

Team

Roll No	Name	Role
CB.EN.P2AID20004	Adithya Babu	
CB.EN.P2AID20018	Devipreetha R	
CB.EN.P2AID20036	PY Sagar	

1.Problem Statement/Objective

Given images of Indian roads. Our aim of this case study is to determine a model that will perform the following:

- Detect 2-Wheeler Vehicle.
- Detect no of passengers in 2-Wheeler.
- Detect helmets in 2-Wheeler.
- Classify a person wearing helmet and not wearing helmet.

2.Dataset Description

The data set has 500 images The dataset has images belonging to 2 classes – One Wearing Helmet, Not Wearing Helmet. The data set has challenges like Size, Location of the object while identifying, Motion Blur of the image.

They are of multiple sizes having different resolution.

It is a RGB color model dataset in JPG or PNG formats.

Data Acquisition is done by collecting Web images and through taking real life images from personal camera.



Sample images



3.Analytical Questions

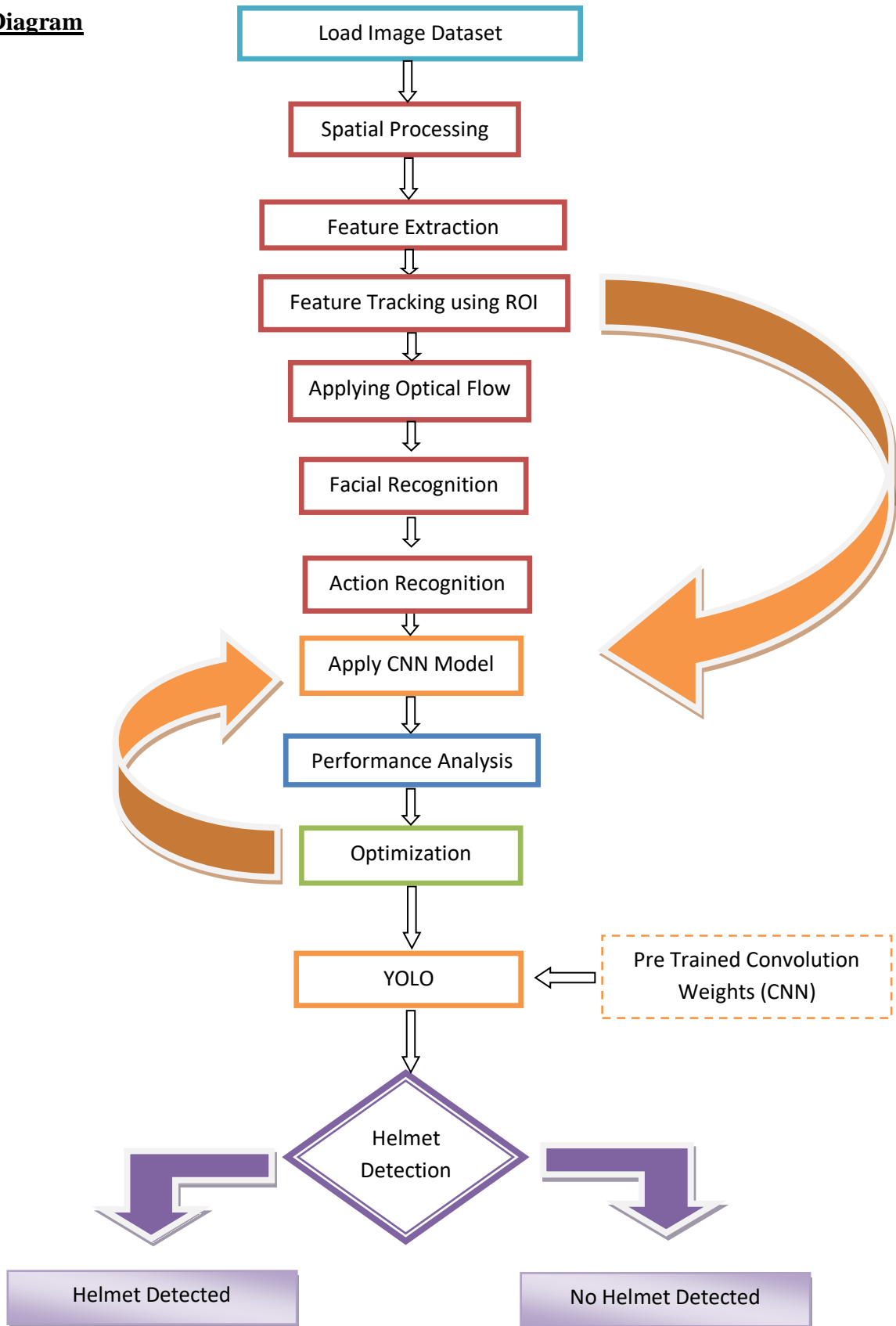
Analysis Questions:

- 1.How many people are wearing helmet?
2. How many single passenger motorcycles are there?
- 3.Are the 2 passenger in motor cycle wearing helmet?
4. Whether only one passenger is wearing helmet?
- 5.How many passengers are there in motor cycle?
6. Which are the locations that people wear helmet?
7. Which are the locations that people does not wear helmet ie-Rural or Urban Streets?
- 8.What is the accuracy of model after this classification?
- 9.Which age group are wearing helmets in roads?
- 10.Which age group are not wearing helmets in roads?

Analytical Questions: [Historical Data]

- 1.What is the average time that person is not wearing helmet?
- 2.Why is a person not wearing helmet?
- 3.Why are both the passenger not wearing helmet?
- 4.Do people wear helmet all the time?
5. Is there any particular time in a day usually tend to remove helmet?
- 6.Do people use helmet in less traffic roads?
- 7.Do teenagers of age 18-24 wear helmet regularly?
- 8.Who usually fails to wear helmets (Men or Women)?
9. Why do people carry helmets in their hands and not wear them?
10. Do people carry their helmets in their arms just to avoid fine?

Block Diagram

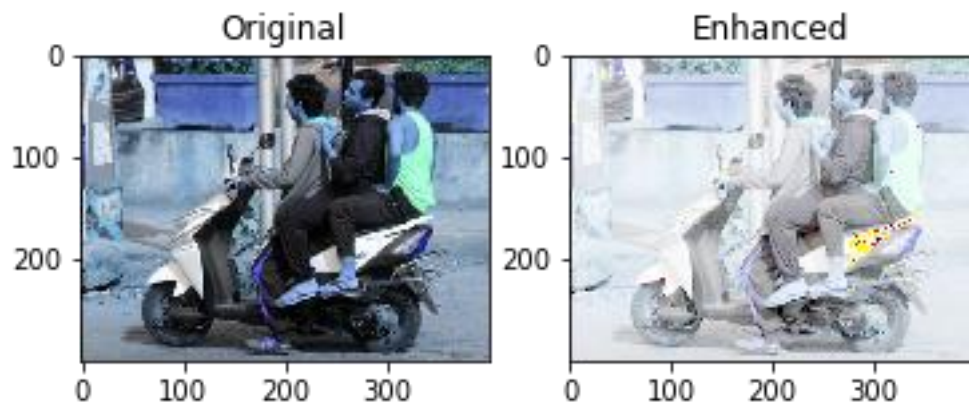


Literature Survey

1.A) Preprocessing In Spatial Domain

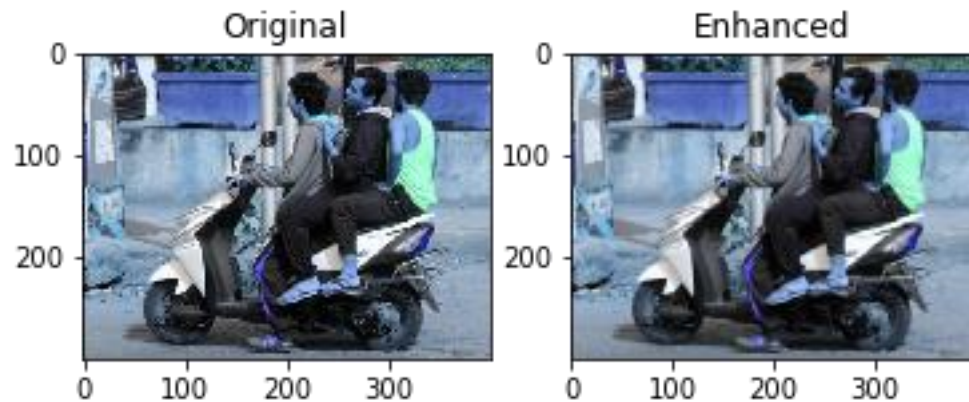
1) Log Transform

The dark pixels in an image are expanded as compared to the higher pixel values. The higher pixel values are kind of compressed in log transformation that is shown in the below image.



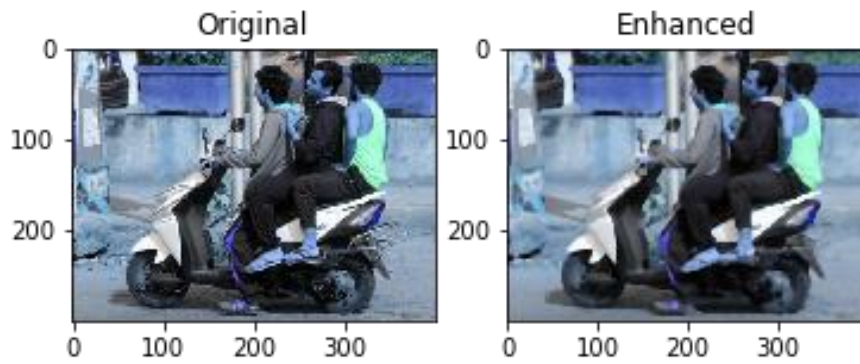
2) Gamma Transform

Gamma transform controls the overall brightness of an image. Images which are not properly corrected can look either bleached out, or too dark



3) Gaussian Blurring

Gaussian blurring is used to show the smoothing of an image by reducing the image noise using Gaussian function.



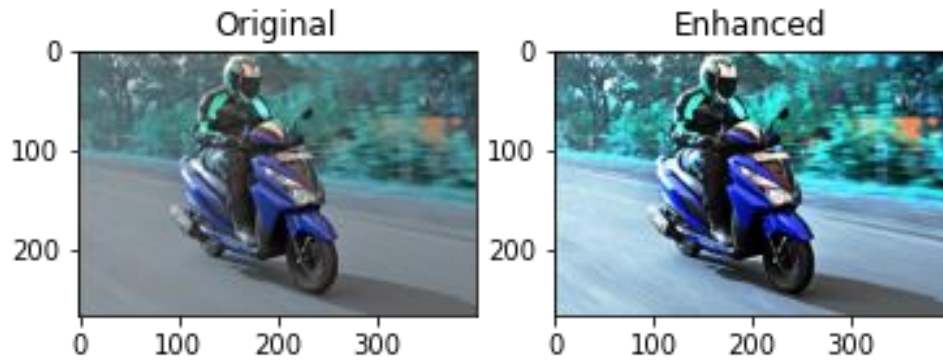
4) Median Filtering

Median filtering is often used to remove noise from an image.



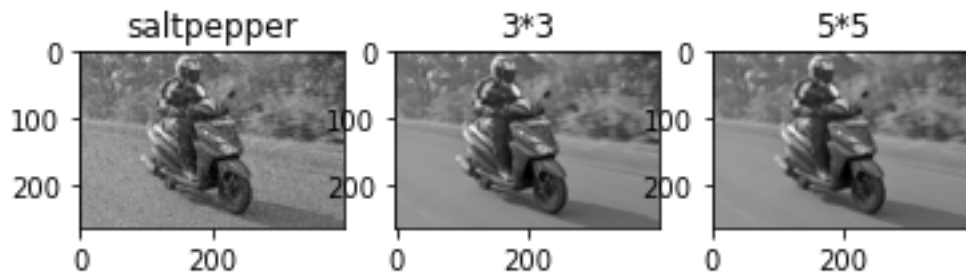
5) Image Sharpening

Image sharpening is an effect applied to images to give them a clearer and sharper appearance.



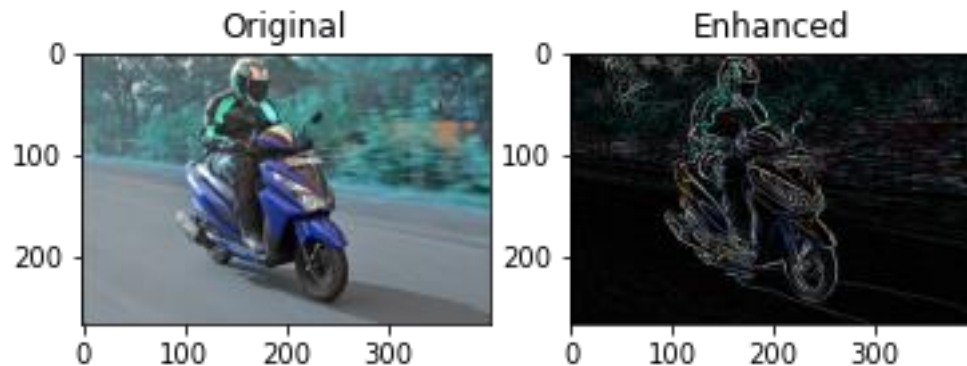
6) Median Blur(With noise)

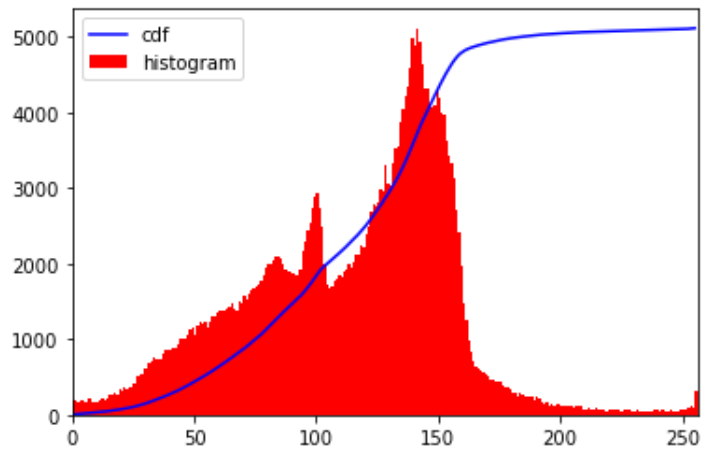
It preserves edges while removing noise.



7) Histogram Equalization

Histogram is a graphical representation of the intensity distribution of an image. In simple terms, it represents the number of pixels for each intensity value considered. It is a method in image processing of contrast adjustment using the image's histogram.

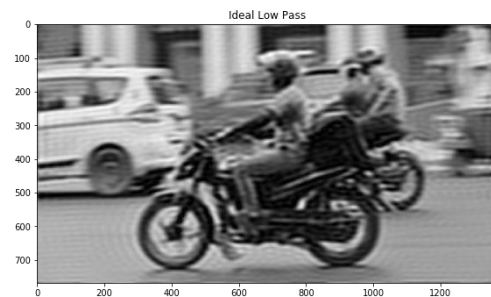
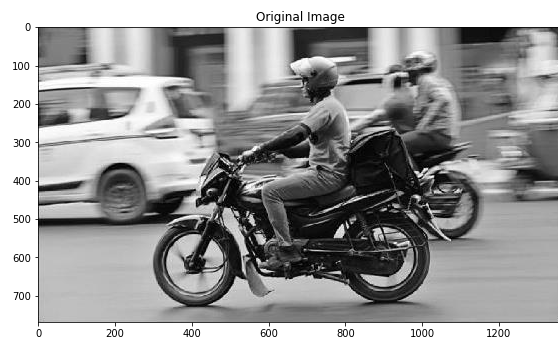




1.B) Image Processing in Frequency Domain

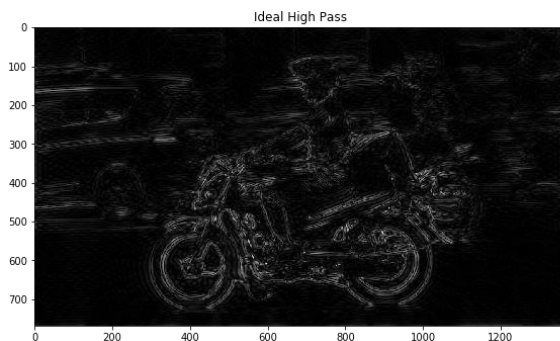
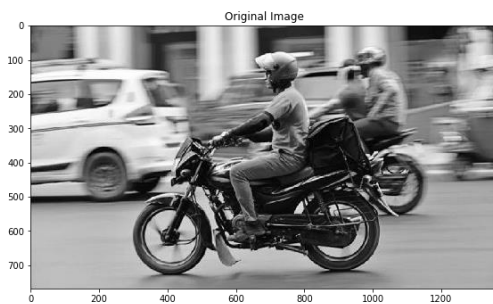
1) Ideal Low Pass

Ideal Low pass Filter (ILPF) is used for image smoothing in the frequency domain. It removes high-frequency noise from a digital image and preserves low-frequency components



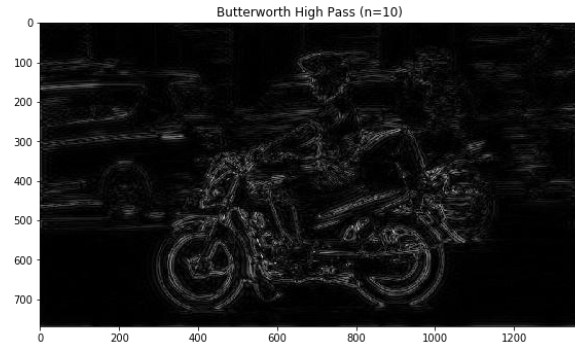
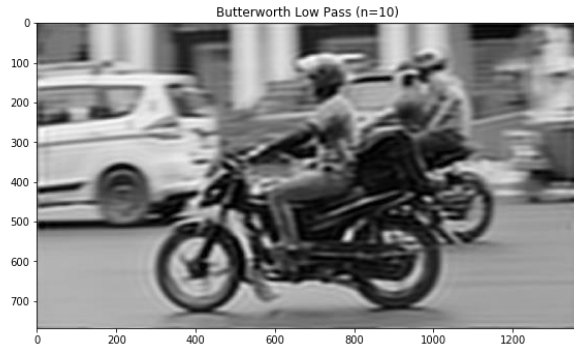
2) Ideal High Pass

Ideal High pass Filter (ILHF) is used for image smoothing in the frequency domain. It removes high-frequency noise from a digital image and preserves high-frequency components.



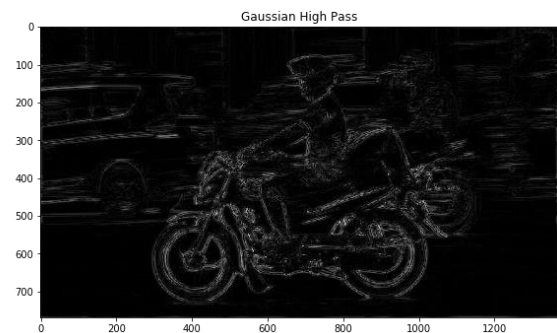
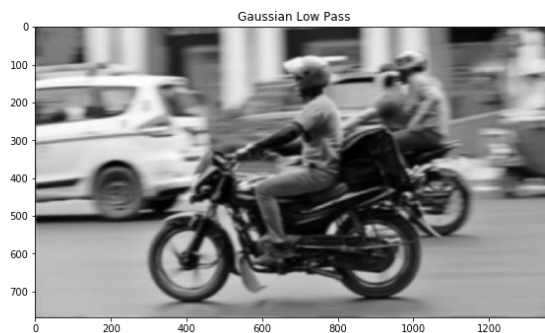
3) Butterworth Filter

Butterworth Filter is used for image sharpening in the frequency domain. Image Sharpening is a technique to enhance the fine details and highlight the edges in a digital image.



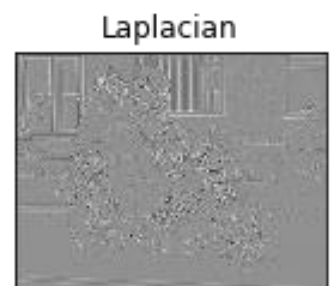
4) Gaussian Filter

Gaussian filter modifies the input signal by convolution with a Gaussian function.



5) Laplacian Filter

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection.

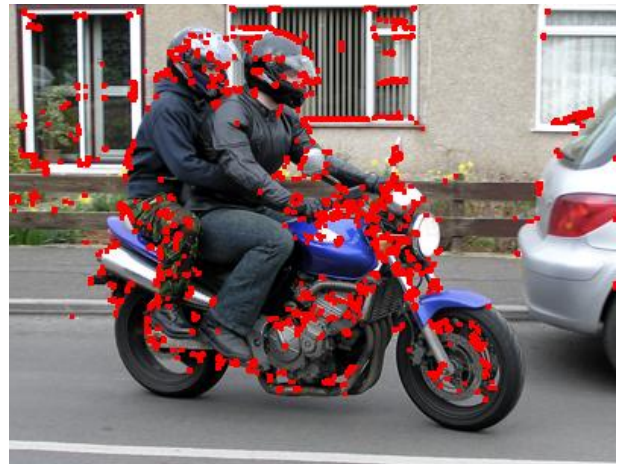


2) Feature Extraction And Tracking

A) Scale-invariant feature transform (SIFT)

Algorithm : This feature detection **algorithm** in computer vision to detect and describe local features in images.

Output:



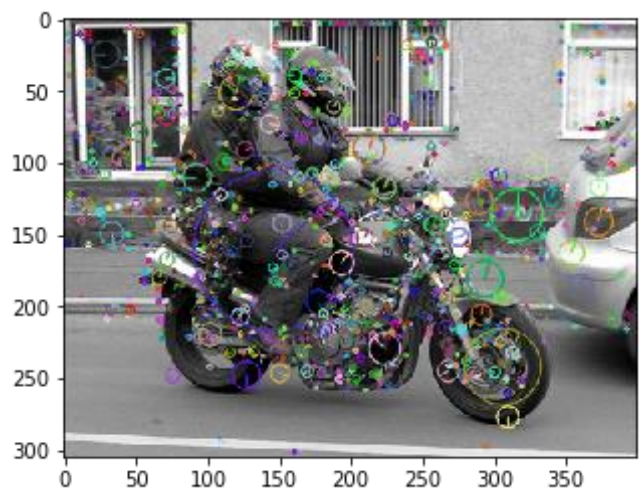
Observation:

The neat edges are detected and highlighted in the image.

B) Harris Corner Detection

Algorithm : Harris Corner Detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image.

Output



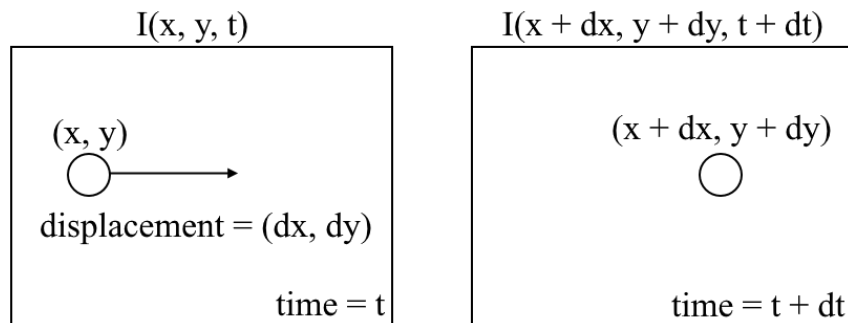
Observation

The corners and boundaries of the features are highlighted in the image.

Features	Purpose	Category
Shape	To detect the helmet	Image
Region	Upper part of image	Image
Texture	Detect variation in light during night and day	Image
Size	Detect the helmet	Image
Area	Area of Helmet	Image

3) Optical Flow

Optical flow is the motion of objects between consecutive frames of sequence, caused by the relative movement between the object and camera. The problem of optical flow may be expressed as:



where between consecutive frames, we can express the image intensity (I) as a function of space (x, y) and time (t) . In other words, if we take the first image $I(x, y, t)$ and move its pixels by (dx, dy) over dt time, we obtain the new image $I(x + dx, y + dy, t + dt)$.

Sparse optical flow gives the flow vectors of some "interesting features" (say few pixels depicting the edges or corners of an object) within the frame while Dense optical flow, which gives the flow vectors of the entire frame (all pixels) - up to one flow vector per pixel. Dense optical flow has higher accuracy at the cost of being slow/computationally expensive.

- **Sparse Optical Flow**

Sparse optical flow selects a sparse feature set of pixels (e.g. interesting features such as edges and corners) to track its velocity vectors (motion). The extracted features are passed in the optical flow function from frame to frame to ensure that the same points are being tracked. There are various implementations of sparse optical flow, including the Lucas–Kanade method, the Horn–Schunck method, the Buxton–Buxton method, and more.

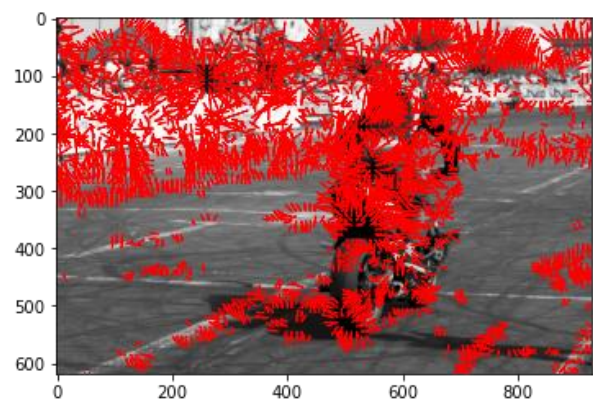
- **Lucas-Kanade Sparse Optical Flow**

Lucas and Kanade proposed an effective technique to estimate the motion of interesting features by comparing two consecutive frames. The Lucas-Kanade method works under the following assumptions:

1. Two consecutive frames are separated by a small time increment (Δt) such that objects are not displaced significantly (in other words, the method works best with slow-moving objects).
2. A frame portrays a “natural” scene with textured objects exhibiting shades of gray that change smoothly.

- **Horn-Schunck Method**

Horn-Schunck method is a classical optical flow estimation algorithm. It assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions which show more smoothness. In this case study, we have implemented the Horn-Schunck method, using smoothing parameter, tolerance and image variation flow vectors for better performance, similar to that of Lucas-Kanade method. Many current optical flow algorithms are built upon its framework.



- **Face Recognition**

Object detection is one of the computer technologies that is connected to image processing and computer vision. It is concerned with detecting instances of an object such as human faces, buildings, trees, cars, etc. The primary aim of face detection algorithms is to determine whether there is any face in an image or not.

Although there are quite advanced face detection algorithms, especially with the introduction of deep learning, the introduction of Viola Jones algorithm in 2001 was a breakthrough in this field. In our Project , we have used Viola Jones Algorithm to detect the faces in the dataset.

- **Viola Jones Algorithm**

Viola Jones algorithm is named after two computer vision researchers who proposed the method in 2001, Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features”. Despite being an outdated framework, Viola-Jones is quite powerful, and its application has proven to be exceptionally notable in real-time face detection. This algorithm is painfully slow to train but can detect faces in real-time with impressive speed.

Given an image(this algorithm works on grayscale image), the algorithm looks at many smaller subregions and tries to find a face by looking for specific features in each subregion. It needs to check many different positions and scales because an image can contain many faces of various sizes. Viola and Jones used Haar-like features to detect faces in this algorithm.

The Viola Jones algorithm has four main steps, which we shall discuss in the sections to follow:

1. **Selecting Haar-like features**

Features like Edge features, Line-features & Four-sided features.

2. **Creating an integral image**

A Summed Up Table for quick and efficient way to calculate the sum of pixel values in an image or rectangular part of an image.

3. **Running AdaBoost training**

So we use the AdaBoost algorithm to identify the best features of the total number of features . AdaBoost checks the performance of all classifiers that you supply to it. So when we're training the AdaBoost to identify important features ,ultimately, the algorithm is setting a minimum threshold to determine whether something can be classified as a useful feature or not.

4. **Creating classifier cascades**

The job of the cascade is to quickly discard non-faces, and avoid wasting precious time and computations. Thus, achieving the speed necessary for real-time face detection.

We have applied Viola Jones Algorithm for better precise face recognition.

The results were :

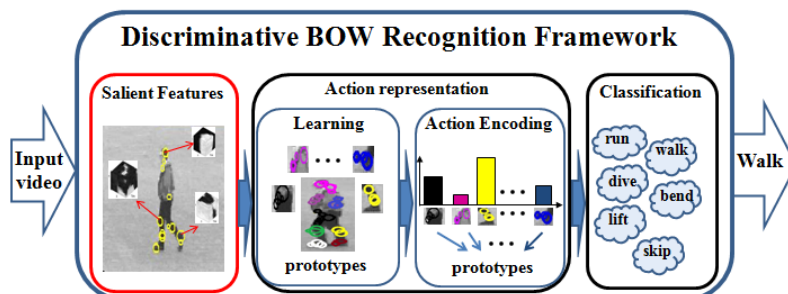


• Action Recognition

Humans easily recognize and identify actions in video but automating this procedure is challenging. Human action recognition in video is of interest for applications such as automated surveillance, elderly behavior monitoring, human-computer interaction, content-based video retrieval, and video summarization

For action recognition we use Spatio-temporal Salient Feature Detection. We have three main steps for the detection. These are feature detection, action representation, and action classification.

From using a bag of words(BOW) or a dataset of action like HMDB51 we can output the specific action.



Action	Frame (from video)
1)_ Movement in Two Wheeler	
2) Wearing a Helmet	
3) Removing a Helmet	
4) Standing in Traffic signal	

- **Deep learning Architectures**

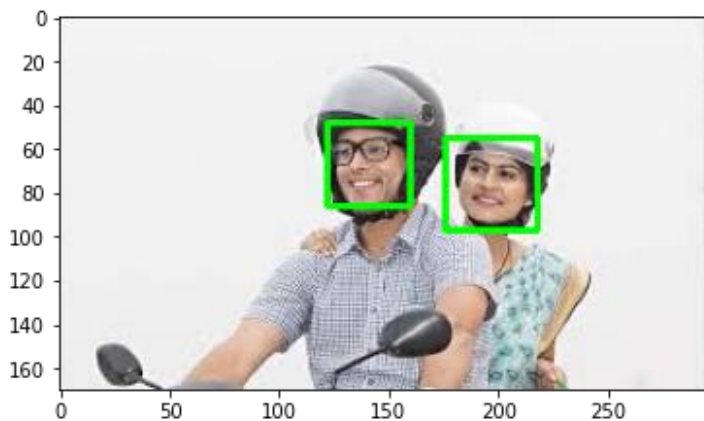
Architecture Name	Category	Learning	Year of Design	Applications
Yolo	DarkNet with fast R-CNN	Supervised	2015	<ul style="list-style-type: none"> • Performs faster detection on various region proposals and thus end up performing prediction multiple times for various regions in an image. • YOLO predicts the coordinates of bounding boxes directly using fully connected layers on top of the convolution feature extractor. Predicting offsets instead of coordinates simplifies the problem and makes it easier for the network to learn.
GoogleNet	Spatial Processing in CNN	Supervised	2015	<ul style="list-style-type: none"> • GoogLeNet architecture was designed for increased computational efficiency with the architecture consisting of 22 layers (27 layers including pooling layers). • GoogLeNet achieves efficiency through reduction of the input image, whilst simultaneously retaining important spatial information.
AlexNet	CNN	Supervised	2012	<ul style="list-style-type: none"> • The architecture consists of eight layers: five convolutional layers and three fully-connected layers. • AlexNet trains the network using graphical processing units(GPUs) , ReLU Nonlinearity and Overlapping pooling.
VGG	Deep CNN	Supervised	2014	<ul style="list-style-type: none"> • VGG consists of 16 weight layers including thirteen convolutional layers with filter size of 3 X 3, and fully-connected layers with filter size of 3 X 3, and the stride and padding of all convolutional layers are fixed to 1 pixel. • The configurations of fully-connected layers in VGG-16 are the same with AlexNet. AlexNet retains more unrelated background information in last convolutional layer, which often disturbs the final prediction. and Hence VGG predicts better than AlexNet.

- **Description of the Scene**

Our Scene would basically consist an image from the roads detecting the 2 wheelers wearing helmet or not.

The background usually would be blurry as the bike would be in motion.

Our objective is to detect the circular shaped helmet on our region of interest (ROI).



9) List of Objects in scene and the features

Objects	Features
Helmet	Shape , Texture, Size
Face	Area , Texture
Two Wheelers	

Methodology

- Object Recognition

1. Yolo (You Only Look Once) Algorithm

YOLO is a real-time object detection algorithm, which is one of the most effective object detection algorithm due to its tremendous speed and accuracy. YOLO came on the computer vision scene with the seminal 2015 paper by Joseph Redmon “You Only Look Once: Unified, Real-Time Object Detection,” then the following year different version of YOLO were published by the same authors till 2018 YOLO v3.

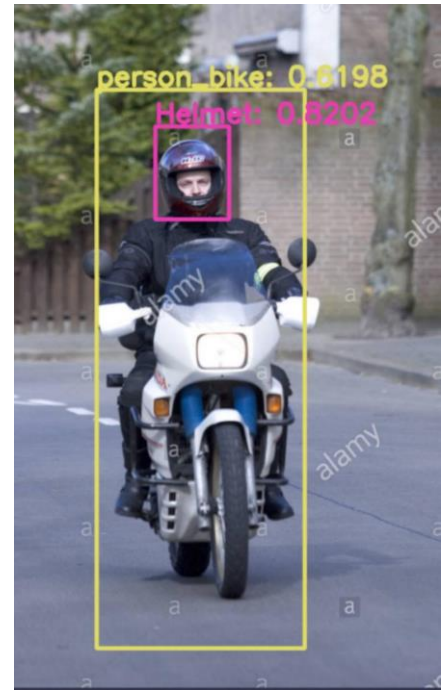
The Version we are going to use is Yolo V3 which has

- 106 layers neural network
- Detection on 3 scales for detecting objects of small to very large size
- 9 anchor boxes taken; 3 per scale. Hence more bounding boxes are predicted than YOLO9000 & YOLOv1
- MultiClass problem turned in MultiLabel problem
- Quite good with small as well as large objects.

Procedure :

- It Intakes an image and divides it in a grid of $S \times S$ (where S is a natural number)
- Each pixel in the image can be responsible for a finite number of (5 in our case) bounding box predictions.
- A pixel is taken responsible for prediction when it is the center of the object detected. Out of all detected boxes, It is taken responsible for the detection of only one object and other detections are rejected.
- For example if the image is divided in a 2×2 grid, and 5 boxes are predicted with 2 classes(Helmet & Face), we will have $2*2(5*5+2)$ predictions=108 predictions.
- Two methods are taken into consideration for selecting final detection
 - **IOU threshold:** The IOU threshold refers to the total intersect area by the union area. If the value is above a threshold value we remove the boundary boxes.
 - **Use Non-Max suppression:** It helps us to avoid duplicate detections of the same object by rejecting multiple predictions for the same object. It takes a list of predicted boxes for the same image and accepts detection depending on the IoU value.

We applied Yolo after giving the pre-trained weights from our deep learning architecture and the output were:



2. Face Detection Algorithms

Face Detection	Dataset	Deep Learning Architecture
Voila Jones	Web Images	Adaboost
Face-Net	Web Images	CNN
Maximum Margin Object Detector(MMOD)	Web Images	CNN
Single Shot Multi Box Detector	Web Images	Res-Net

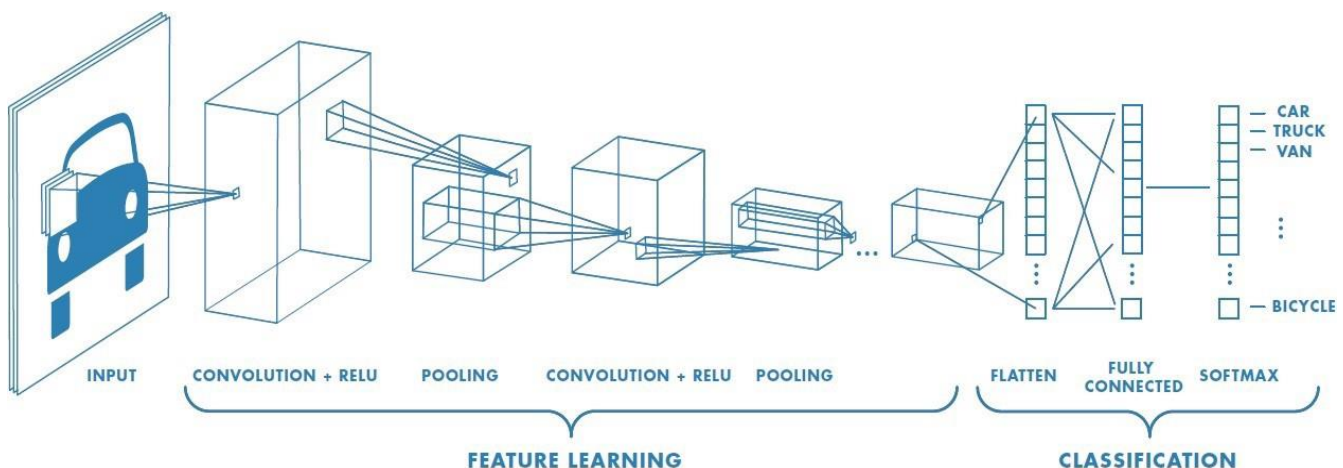
- **Deep learning architecture**

- 1) **Convolutional Neural Networks**

A **Convolution Neural Network (CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics.

The architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

A CNN is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

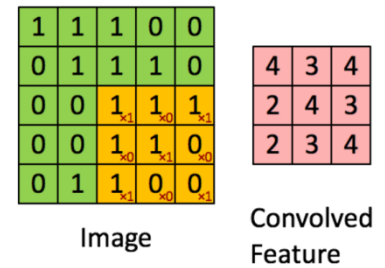


Convolutional Neural Networks have the following layers:

- Convolution layer
- Pooling layer
- Fully Connected Layer

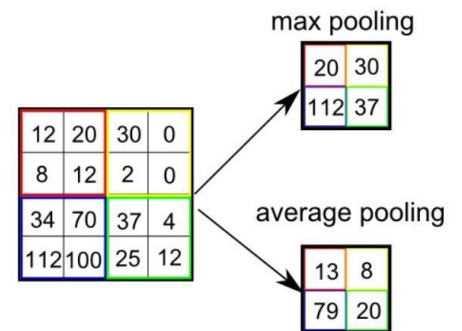
Convolution layer

We will transfer the features to every other position of the image and will see how the features match that area completely and add more convolution layers to detect more features.



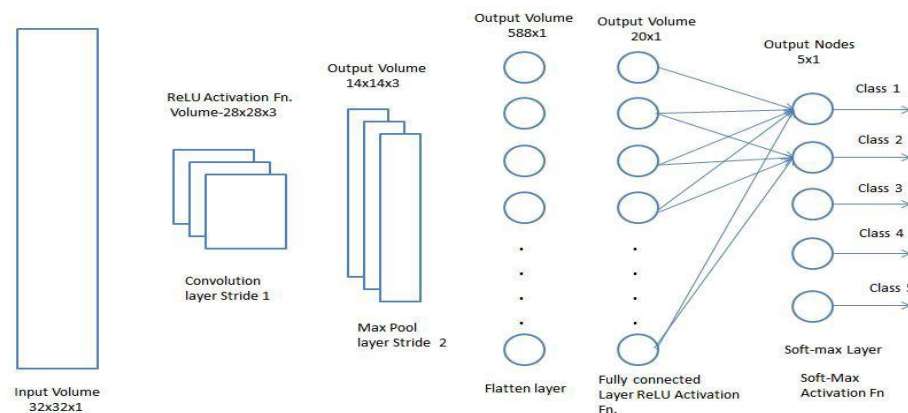
Pooling layer

The main role in this layer is to process the image generated after convolution, reduce the size of the model, increase the calculation speed, and improve the robustness of the extracted features. There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.



Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.



- **Parameters and Hyper-parameters**

Parameters	Hyperparameters – Expected value
<p>1. Input Layer All the input layer does is read the image.</p> <p>2. Convolution Layer This layer converts all the pixels in its receptive field into a single value. For example, if you would apply a convolution to an image, you will be decreasing the image size as well as bringing all the information in the field together into a single pixel. The final output of the convolution layer is a vector. A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image.</p> <p>3. Pooling Layer A pooling layer is another building block of a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer.</p>	<p>1. Kernel Size –the size of the filter.</p> <p>2. Kernel Type–values of the actual filter (e.g., edge detection, sharpen).</p> <p>3. Stride–the rate at which the kernel pass over the input image. Value is 1 (Default)</p> <p>4. Padding–add layers of 0s to make sure the kernel pass over the edge of the image.</p> <p>5. Hidden layer–layers between input and output layers.</p> <p>6. Activation functions–allow the model to learn nonlinear prediction boundaries. Functions are :ReLU , sigmoid.</p> <p>7. Learning rate–regulates on the update of the weight at the end of each batch. Value ranges from 0.001</p> <p>8. Momentum–regulates the value to let the previous update influence the current weight update.</p> <p>9. Epochs–the iterations of the entire training dataset to the network during training. Values usually are 25 (Default) [100 – 1000]</p> <p>10. Batch size–the number of patterns shown to the network before the weights are updated. Models can have more than 10 hyper parameters and finding the best combination can be view as the search problem. Values from 50.</p>

▪ Parameters applied for our Project


Inputs/Weights	Inputs/Weights
<p>Convolution Layers = 3 MaxPooling Layers = 3 Dense Layers = 2 Flatten Layers = 2</p> <p>Output- Object is correctly identified</p>	<p>The right choice of hyperparameter values can affect the performance of the model.</p> <ol style="list-style-type: none"> 1. Kernel Size – 3*3 2. Kernel Type – Depending on the 3. Stride – 1 (Default) 4. Padding 5. Hidden layer 6. Activation functions –ReLU , sigmoid. 7. Learning rate- 0.002(float) 8. Epochs- 25 (Default) [100 – 1000] 9. Batch size-50 (Integer) <p>Output- Enhanced and Optimized Object Identification</p>

• **Training , Testing & Validation.**

- The sample of data used to fit the model is usually the **Training Dataset**. The actual dataset that we use to train the model (weights and biases in the case of a Neural Network). The model sees and learns from this data. While splitting the dataset the major proportion would belong to training dataset (70-80) .
- The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset is called the **Test Dataset**. The Test dataset provides the gold standard used to evaluate the model. It is only used once a model is completely trained.(10% is used as test dataset)
- The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters is called the **Validation Dataset** The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model and use this data to fine-tune the model hyper parameters.(10% is used as validation dataset)

- Performance Metrics

Metric	Category	Purpose	Formula
Peak signal-to-noise ratio (PSNR)	Image Enhancement	Analysis of Quality of Image	$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{dB}$
Mean Squared Error (MSE)	Image Enhancement	Analysis of Quality of Image	$MSE = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [X(i,j) - Y(i,j)]^2$
False Positive Rate	Face Detection	For checking the accuracy of Viola Jones Algorithm	$\text{false positive rate} = \frac{\text{false positives}}{\text{false positives} + \text{true negatives}}$

Intersection over Union (IoU)	Object Identification	To check for true positives and false positives based on the overlapping area of the boundary boxes.	 $IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$
Mean Average Precision (mAP)	Object Identification	For evaluating the localization performance using precision-recall curve	$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$ <p>$AP_k = \text{the AP of class } k$ $n = \text{the number of classes}$</p>

References

- **Image Dataset – [Web Images](#) & Camera**
- **Deep Learning – CNN**
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- **Yolo Architecture**
<https://medium.com/@ODSC/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0>
- **Research Papers :**
 - [Image Captioning with Object Detection and Localization](#)
 - [Hyperparameter Optimization in Convolutional Neural Network](#)
 - [Fine Tuning of Convolution Neural Network](#)
 - [Detection and analysis of wheat spikes using Convolutional Neural Networks](#)
 - [Deep Learning-Based Safety Helmet Detection in Engineering Management](#)