

# Facial Recognition in Crime Scene

Vasu Upadhayay

Department of Information Technology  
PSG College of Technology  
Coimbatore, India

**Abstract:-** Whenever any crime occurs, the video footage of that area is analysed by detectives or other agencies manually to nab criminals. This consumes a lot of time and effort, thus paving the way for the criminal to escape. "Facial recognition in crime scene" shows a list of probable criminals who have passed through that scene, thus minimizing manual search time and effort. A recent report claimed that crimes in India saw a marginal increase in the first 45 days of this year as compared to the corresponding period of the previous year. So, intelligence agencies can save a lot of time and manpower by employing this product. Our intended or target audience includes police departments, intelligence agencies, and bureaus and also any other private detective organizations. Our model involves converting video to images or frames. This has been done by the use of OpenCV. OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer algorithms including image and video processing algorithms that have been used for processing the crime scene footage. Objects and humans are detected in each frame and stored using a pre-trained model RetinaNet. RetinaNet is a pre-trained model that can classify objects and humans accurately with the label. In RetinaNet, a one-stage detector, by using focal loss, a lower loss is contributed by easy negative samples so that the loss is focusing on hard samples, which improves the prediction accuracy. Obtained human images are sent to a MATLAB that is used to run a facial recognition code across a database. This database is the organization's employee database and so we can tell who are the employees of that organization who crossed the crime scene during the specified time. So we can get a separate list as employees and others.

**Key Words:** *RetinaNet, OpenCV, Focal loss, Footage, MATLAB*

## I. INTRODUCTION

A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected facial features from given image with faces within a database. It is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analysing patterns based on the person's facial textures and shape. While initially a form of computer application, it has seen wider uses in recent times on mobile platforms and in other forms of technology, such as robotics. It is typically used as access control in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Although the accuracy of facial recognition system as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless and non-invasive process. Recently, it has also become popular as

a commercial identification and marketing tool. Other applications include advanced human-computer interaction, video surveillance, automatic indexing of images, and video database, among others.

### 1.1. Motivation

A recent report claimed that crimes in India saw a marginal increase in the first 45 days of this year as compared to the corresponding period of the previous year and analyzing the video footage is one of the important parts in tracing the criminal.

Whenever any crime occurs, the video footage of that area is analyzed by detectives or other agencies manually to nab criminals. This consumes a lot of time and effort, thus paving the way for the criminal to escape. "Facial recognition in crime scene" shows a list of probable criminals who have passed through that scene, thus minimizing manual search time and effort. So, intelligence agencies can save a lot of time and manpower by employing this product.

### 1.2. OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. OpenCV is written in C++ and in

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older

C interface. There are bindings in Python, Java and MATLAB/OCTAVE. OpenCV has a modular structure, which means that the package includes several shared or static libraries. The module used in this project is the Video Analysis module.

### 1.3. RETINANET

In RetinaNet, a one-stage detector, by using focal loss, the lower loss is contributed by "easy" negative samples so that the loss is focusing on "hard" samples, which improves the prediction accuracy. With ResNet+FPN as the backbone for feature extraction, plus two task-specific subnetworks for classification and bounding box regression, forming the RetinaNet, which achieves state-of-the-art performance, outperforms Faster R-CNN, the well-known two-stage detectors. RetinaNet is used for deep feature extraction. Feature Pyramid Network (FPN) is used on top of RetinaNet for constructing a rich multi-scale feature pyramid from one single resolution input image.

## II. LITERATURE REVIEW

### 2.1. Focal loss for dense object detection

#### 2.1.1 Existing System

The highest accuracy object detectors to date are based on a two-stage approach popularized by R-CNN, where a classifier is applied to a sparse set of candidate object locations. In contrast, one-stage detectors that are applied over a regular, dense sampling of possible object locations have the potential to be faster and simpler, but have trailed the accuracy of two-stage detectors thus far. In this paper, we investigate why this is the case.

##### 2.1.1.1 Classic Object Detector

The sliding-window paradigm, in which a classifier is applied on a dense image grid, has a long and rich history. One of the earliest successes is the classic work of LeCun et al who applied convolutional neural networks to handwritten digit recognition. Viola and Jones used boosted object detectors for face detection, leading to widespread adoption of such models. The introduction of HOG and integral channel features gave rise to effective methods for pedestrian detection. DPMs helped extend dense detectors to more general object categories and had top results on PASCAL for many years. While the sliding-window approach was the leading detection paradigm in classic computer vision, with the resurgence of deep learning, two-stage detectors, described next, quickly came to dominate object detection.

##### 2.1.1.2 Two-Stage Detector

The dominant paradigm in modern object detection is based on a two-stage approach. As pioneered in the Selective Search work, the first stage generates a sparse set of candidate proposals that should contain all objects while filtering out the majority of negative locations, and the second stage classifies the proposals into foreground classes/background. R-CNN upgraded the second-stage classifier to a convolutional network yielding large gains in accuracy and ushering in the modern era of object detection. R-CNN was improved over the years, both in terms of speed and by using learned object proposals. Region Proposal Networks (RPN) integrated proposal generation with the second-stage classifier into a single convolution network, forming the Faster RCNN framework. Numerous extensions to this framework have been proposed.

##### 2.1.2 Focal Loss

The Focal Loss is designed to address the one-stage object detection scenario in which there is an extreme imbalance between foreground and background classes during training. We introduce the focal loss starting from the cross-entropy (CE) loss for binary classification:

$$CE(p,y) = \begin{cases} -\log(p) & \text{if } y=1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

In the above  $y \in \{\pm 1\}$  specifies the ground-truth class and  $p \in [0,1]$  is the model's estimated probability for the class with label  $y = 1$ . For notational convenience, we define  $pt$ :

$$pt = p \text{ if } y = 1$$

$1-p$  otherwise,

and rewrite  $CE(p,y) = CE(pt) = -\log(pt)$ .

The CE loss can be seen as the blue (top) curve in Figure 1. One notable property of this loss, which can be easily seen in its plot, is that even examples that are easily classified incur a loss with non-trivial magnitude. When summed over a large number of easy examples, these small loss values can overwhelm the rare class.

##### 2.1.3 Balanced Entropy

A common method for addressing class imbalance is to introduce a weighting factor  $\alpha \in [0,1]$  for class 1 and  $1-\alpha$  for class -1. In practice,  $\alpha$  may be set by inverse class frequency or treated as a hyperparameter to set by cross-validation. For notational convenience, we define  $\alpha t$  analogously to how we defined  $pt$ . We write the  $\alpha$ -balanced CE loss as

$$CE(\alpha t) = -\alpha t \log(\alpha t)$$

This loss is a simple extension to CE that we consider as an experimental baseline for our proposed focal loss.

##### 2.1.4 Class Imbalanced and Two-Stage Detectors

Two-stage detectors are often trained with the cross-entropy loss without the use of  $\alpha$ -balancing or our proposed loss. Instead, they address class imbalance through two mechanisms: a two-stage cascade and biased minibatch sampling. The first cascade stage is an object proposal mechanism that reduces the nearly infinite set of possible object locations down to one or two thousand. Importantly, the selected proposals are not random but are likely to correspond to true object locations, which removes the vast majority of easy negatives. When training the second stage, biased sampling is typically used to construct mini-batches that contain, for instance, a 1:3 ratio of positive to negative examples. This ratio is like an implicit  $\alpha$  balancing factor that is implemented via sampling. Our proposed focal loss is designed to address these mechanisms in a one-stage detection system directly via the loss function.

### 2.2. Feature Pyramid Networks For Object Detection

#### 2.2.1 Existing System

One stage object detection is used for detecting objects.

##### 2.2.1.1 Feature Pyramid Network Backbone

We adopt the Feature Pyramid Network (FPN) from as the backbone network for RetinaNet. In brief, FPN augments a standard convolutional network with a top-down pathway and lateral connections so the network efficiently constructs a rich, multi-scale feature pyramid from a single resolution input image. Each level of the pyramid can be used for detecting objects at a different scale. FPN improves multi-scale predictions from fully convolutional networks (FCN), as shown by its gains for RPN and Deep Mask-style proposals, as well as two-stage detectors such as Fast R-CNN or Mask R-CNN.

##### 2.2.1.2 RetinaNet Detector

RetinaNet is a single, unified network composed of a backbone network and two task-specific subnetworks. The backbone is responsible for computing a convolutional

feature map over an entire input image and is an off-the-self convolutional network. The first subnet performs convolutional object classification on the backbone's output; the second subnet performs convolutional bounding box regression. The two subnetworks feature a simple design that we propose specifically for one-stage, dense detection. While there are many possible choices for the details of these components, most design parameters are not particularly sensitive to exact values as shown in the experiments. We describe each component of RetinaNet next.

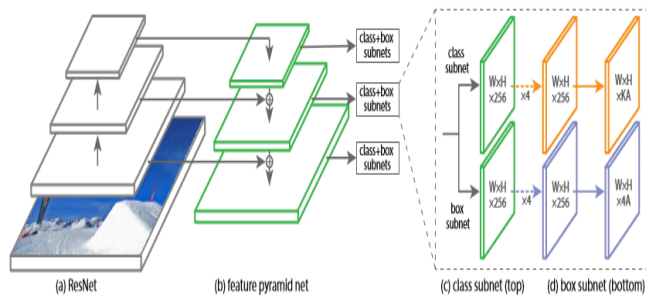


Fig 2.1 Focal Loss Object Loss

### III. PROPOSED SYSTEM

Our proposed system can be divided into three major modules. One being the frame generation part and one being object detection and the last facial recognition.

#### 3.1. Frame Generation

The problem requires that we split the input video into multiple frames. The video is a collection of frames is split into  $n$  number of frames for the further steps that follow the step of frame generation. This frame could thus be treated as an image that can be worked upon for facial recognition or even feature extraction. This process of converting a video to images makes way for the use of image processing algorithms. In our project, we have used packages and libraries to help with our problem statement. The library we have chosen is the OpenCV in python. This part is done with the use of OpenCV. OpenCV (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision.

#### 3.2. Object Recognition

Once we have frames or images that have been generated from the video, the next step involves the part where object recognition is done. This is also done by keeping in mind that the number of frames that have been extracted is too large in number. There would be frames where there are no objects like humans. These are certain restrictions or points to be noted. RetinaNet is one of the most widely used and most preferred models for object detection. It is a single, unified network composed of a backbone network and two task-specific subnetworks. The backbone is responsible for computing a feature map over an entire input image and is an off-the-self convolution network. The first subnet performs classification on the backbone's output; the second subnet performs convolution bounding box regression.

This model is pre-trained. So we just have to download it to generate predictions. We have also used the image library. This is a python library that supports state-of-the-art machine learning algorithms for computer vision tasks. Using this image has even eliminated the need for complex installation scripts or even GPU. The output predictions are directly got through the use of this library.

#### 3.3. Facial Recognition

Face recognition leverages computer vision to extract discriminative information from facial images, and pattern recognition or machine learning techniques to model the appearance of faces and to classify them. It uses computer vision techniques to perform feature extraction to encode the discriminative information required for face recognition as a compact feature vector using techniques and algorithms such as:

- Dense local feature extraction with SURF, BRISK or FREAK descriptors.
- Histogram of oriented gradients.
- Distance between detected facial landmarks such as eyes, noses, and lips.
- Machine Learning techniques that can be applied to the extracted features to perform face recognition or classification using:
  - Supervised Learning techniques such as support vector machines (SVM) and decision trees.
  - Ensemble learning methods.
  - Deep neural networks.

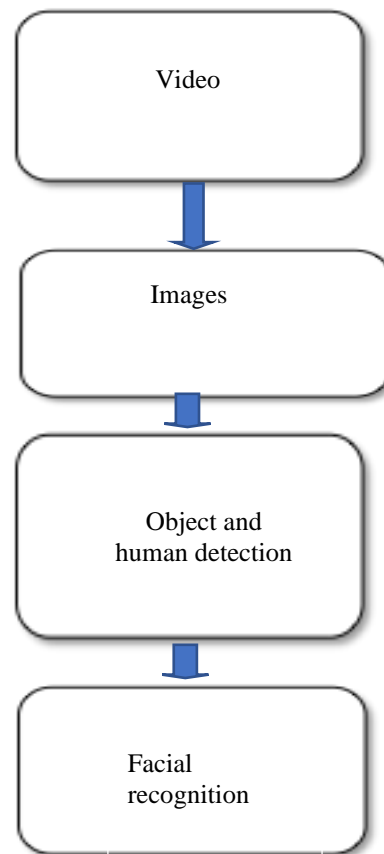


Fig -3.1 Steps in our workflow

#### IV. DESIGN AND IMPLEMENTATION

#### V. CONCLUSION

##### 4.1. Video To Frame Conversion

OpenCV comes with many powerful video editing functions. In the current scenario, techniques such as image scanning, face recognition can be accomplished using OpenCV. OpenCV library can be used to perform multiple operations on videos. Let's try to do something interesting using CV2. Take a video as input and break the video into a frame by frame and save those frames. Now, the number of operations can be performed on these frames. Like reversing the video file or crop the video etc. For playing video in reverse mode, we need only to store the frames in a list and iterate reverse in the list of frames. Use the reverse method of the list for reversing the order of frames in the list.

##### 4.2. Object And Person Detection

When we're shown an image, our brain instantly recognizes the objects contained in it. On the other hand, it takes a lot of time and training data for a machine to identify these objects. But with the recent advances in hardware and deep learning, this computer vision field has become a whole lot easier and more intuitive.

Object detection technology has seen a rapid adoption rate in various and diverse industries. It helps self-driving cars safely navigate through traffic, spots violent behavior in a crowded place, assists sports teams to analyze and build scouting reports, ensures proper quality control of parts in manufacturing, among many, many other things.



Fig -4.1 Detecting humans

Facial Recognition in a crime scene model can be used to filter out probable criminals by analyzing a particular time frame in video footage. The obtained criminal image can be then used by government officials to catch the criminal. This helps in saving time, energy and helps to catch the criminal on time with less effort. This also reduces the crime rate as we can catch the criminal faster.

#### VI. REFERENCES

- [1] Tsung-Yi Lin<sup>1,2</sup>, Piotr Dollár<sup>1</sup>, Ross Girshick<sup>1</sup>, Kaiming He<sup>1</sup>, Bharath Hariharan<sup>1</sup>, and Serge Belongie<sup>2</sup>, "Feature Pyramid Networks for Object Detection", IEEE Conference on Computer Vision and Pattern Recognition, 2017
- [2] Tsung-Yi Lin<sup>1</sup>, Piotr Dollar, Kaiming He, Ross Girshick Priya Goyal, "Focal Loss for Dense Object Detection", IEEE Conference on Computer Vision and Pattern Recognition, 2017
- [3] Himashu Chauhan, and Dr. Sandhya Tarar, "Image and Video Processing Edge Detection Technique used for Traffic Control Problem," International Journal of Computer Science Trends and Technology (IJCT) – Volume 4 Issue 1, Jan - Feb 2016