Intelligent Video Surveillance System for Improving Safety and Security

Harnoor Singh
Department of ECE
University of Waterloo
Waterloo, Canada
Email: h287sing@uwaterloo.ca

Jubilee Imhanzenobe
Department of ECE
University of Waterloo
Waterloo, Canada
Email: jimhanze@uwaterloo.ca

Ishpinder Kaur
Department of ECE
University of Waterloo
Waterloo, Canada
Email: 17kaur@uwaterloo.ca

***Abstract* – In recent years, a considerable number of cameras have been put in public places as part of Intelligent Video Surveillance Systems. Human observation of these activities is a difficult, time-consuming task and not always efficient. Recent advancements in sensor development, embedded systems, wireless networks, computer vision and deep learning have allowed us to perform video analytics on such scenes and data captured through the images and videos, which helps in decision making and provide security through Intelligent Video Surveillance, in various sensitive fields without human intervention. In this paper we review the various solutions for employing features of Intelligent Video Surveillance systems like motion detection, object detection, face recognition and crowd detection that have been proposed and used over the years. In recent years, a considerable number of cameras have been put in public places as part of Intelligent Video Surveillance Systems. Human observation of these activities is a difficult, time-consuming task and not always efficient. Recent advancements in sensor development, embedded systems, wireless networks, computer vision and deep learning have allowed us to perform video analytics on such scenes and data , captured through the images and videos, which helps in decision making and provide security through Intelligent Video Surveillance, in various sensitive fields without human intervention. In this paper we review the various solutions for employing features of Intelligent Video Surveillance systems like motion detection, object detection, face recognition and crowd detection that have been proposed and used over the years.**

***Keywords* – object detection, face recognition, deep learning, MTCNN, Haarcascades, YOLO, motion detection**

1. INTRODUCTION

With the emergence of IOT, AI and other technological advancements, every task can be accomplished with a single swipe and click, from any part of the world, using advanced gadgets. This also reflects a picture how huge amount of data is added every second, which needs to be analysed and put it together for other security or prediction purposes. Recent advancements in sensor development, embedded systems, wireless networks, and computer vision have heightened interest in related applications of such enabling technologies in several sectors of Ambient Intelligence. Various incidents or crimes can be prevented using this data especially, captured through the images and videos, but human observation of these activities is a difficult, time consuming and not always efficient. Therefore, analytics can be performed on such scenes and data, where video data can be gathered, monitored, helps in decision making and provide security through Intelligent Video Surveillance, in various sensitive fields without Human intervention. Intelligence Video Surveillance (IVS) also known as surveillance through video analytics, is a crucial part in dealing with real world problems. Video analytics has plethora of applications and been deployed in fields like health sectors, traffic control, oil, and gas mining etc. As of today's need, the CCTVs are deployed around every corner, the social media, thermal cameras etc., which acts as source for this information. Past recorded data can be used for mining insights and future decision making, whereas present data displays the motion patterns of any object and also identifies it.

It's difficult to pinpoint the exact year IVS was born. The first publications on this topic were published in 1996–1998. However, the publishing of a special section on 'Video Surveillance' in IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE T-PAMI journal in August 2000) represents a significant milestone. At the time, it was argued that existing video surveillance systems were primarily used for offline forensic analysis, and that there was a pressing need for real-time, online automated analysis of video streams to alert security officers to a burglary in progress or suspicious behaviour while the crime could still be prevented [1].

Initially, most of the works in IVS focused on the single fixed camera scenario, which was also due to cost and availability constraints. The scene was automatically analysed using the video feed given by a single fixed camera. The single fixed viewpoint offered by a single camera, on the other hand, has some limits in terms of both the quantity of area covered and the robustness against occluding objects. The former constraint precludes the system from monitoring broad regions (as is common in security applications) and from detecting and tracking things that move out of the camera's field of vision. The latter constraint, on the other hand, renders the system unreliable when the targets are (partially) obscured by other moving or stationary objects. To address these restrictions, numerous IVS projects began investigating various types of cameras Mobile Vision and Egocentric cameras were introduced. Mobile vision shares and builds upon prior work on active vision, which deals with computer vision algorithms for processing imagery captured by non-stationary camera around 2002. PTZ (Pan-Tilt-Zoom) or active cameras, for example, have been used to dynamically adjust the viewpoint of a single fixed camera to eliminate occlusions and increase the monitored area. Later on, multiple cameras,

whereas egocentric cameras provide a 'first-person' perspective of the scene.

The algorithm progresses in IVS have followed a low-to-high-level path, similar to other branches of computer vision (and computer science in general). Since, at the beginning, single fixed cameras were used, the first efforts can be classified as the development of effective (and efficient) background suppression techniques. Because single fixed cameras were utilized in the beginning, the early efforts can be characterized as the creation of effective (and efficient) background suppression techniques. Background suppression refers to an algorithm (or set of algorithms) aiming at separating (at pixel- or object-level) the stationary parts of the scene (i.e., the background) from those which are moving (i.e., the foreground). Once the (moving) objects in the scene have been detected, the next step typically involves tracking objects, i.e., consistently associating the same identifier to an object, over time. As a natural extension, once the objects are tracked in each single camera of a network of multiple cameras, there has been a significant number of papers addressing the multicamera object tracking problem.

Thanks largely to advances in Deep Learning research and increased availability of video data with the expansion of global video camera networks, video analytics has transitioned from traditional algorithms based purely on Computer Vision to incorporating powerful Deep Learning techniques. Deep Learning, a subset of Artificial Intelligence, is a training convention by which a machine is exposed to volumes of tagged data in order to "learn" to recognize and identify the same information in new data sets. Imitating the way that a human is taught, Deep Learning enables technologies to detect and identify objects based on increased exposure to information more proficiently. Driven by robust hardware infrastructure, Deep Learning enables faster analytic output, improved processing performance and increased object detection, classification, and recognition accuracy.

2. RELATED WORK

2.1 Motion Detection

It is detecting any change in the position or movement of object or person with respect to its surroundings or vice versa. Video Motion Detection (VMD) was introduced in the early 1990s as a novel method of identifying when pixels in a scene changed. Simple video analytics, built on top of VMD, were released around the year 2000, and covered height and width ratios, object speed, and repetitive motion. The use of algorithms and filters for what prompted an alarm contributed to a reduction in false alarms over VMD.

Background subtraction is a popular method for recognizing moving objects in films captured by static cameras. The approach is based on recognizing moving objects by comparing the current frame to a reference frame, which is commonly referred to as a "background image" or "background model".

2.2 Object Detection

Object detection consists of recognizing, identifying, and locating objects within a picture with a given degree of confidence. An important task in object recognition is to identify what is in the image and with what level of confidence.

2.2.1 Traditional Era of Object Detection

Most early object detection algorithms were built on handcrafted features due to the lack of good image representation at the time.

2.2.1.1 Viola Jones Detectors: This object recognition framework, created by Paul Viola and Michael Jones in 2001, allows for the real-time detection of human faces [10]. It uses sliding windows to examine all potential locations and scales in a picture to determine whether any window contains a human face. The sliding windows essentially searches for 'haar-like' features. As a result, the haar wavelet is employed as an image's feature representation. It leverages integral image to speed up detection by making the computing complexity of each sliding window independent of its window size [11]. Another technique utilized by the authors to increase detection performance is the use of the Adaboost algorithm for feature selection, which picks a limited collection of features that are generally useful for face detection from a large pool of random data. The system also uses Detection Cascades, a multi-stage detection paradigm, to reduce its computational overhead by focusing on face targets rather than backdrop windows [10].

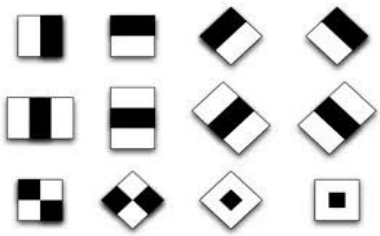

Figure 1: Haar-Like Features

2.2.1.2 HOG Detector: HOG, which was first proposed in 2005 by N. Dalal and B. Triggs [4], is an enhancement to the scale invariant feature transform [5, 6] and shape contexts of its time. HOG employs blocks (similar to a sliding window), a dense pixel grid in which gradients are composed of the magnitude and direction of change in the brightness of pixels within the block. HOGs are well-known for their application in pedestrian detection. The HOG detector rescales the input image numerous times while keeping the size of a detection window constant to detect objects of varying sizes.

2.2.1.3 Deformable part-based model (DPM): P. Felzenszwalb proposed DPM in 2008 as an expansion of the HOG detector [7]. R. Girshick afterwards added a variety of enhancements [8, 9]. DPM uses the strategy that the problem of detecting a "car" can be broken down as a 'divide and conquer' strategy by detecting its window, body, and wheels. A DPM detector is made up of a root-filter and several part-filters. In DPM, a weakly supervised learning approach is developed in which all part filter configurations (size, location, etc.) can be automatically learned as latent variables. R. Girshick used a special example of multi-Instance learning as well as various other essential techniques to boost detection accuracy, including "hard negative mining", "bounding box regression", and "context priming". Later, he used an approach that uses a cascade

architecture to achieve over 10 times acceleration without sacrificing accuracy.

2.2.2 Deep Learning Era

Unfortunately, around 2010, the performance of hand-crafted features hit a saturation point, resulting in a plateau in object detection. However, the world witnessed the resuscitation of convolutional neural networks in 2012, as deep convolutional networks were successful in learning robust and high-level feature representations of an image. The concept of Regions with CNN features (RCNN) for object detection broke the object detection deadlocks in 2014. Object detection is divided into two categories in current deep learning era: "two-stage detection" and "one-stage detection".

2.2.2.1. CNN BASED TWO-STAGE DETECTORS:

a) RCNN: It starts with the extraction of a set of object proposals (object candidate boxes) by selective search. Then each proposal is rescaled to a fixed size image and fed into a pre-trained CNN model to extract features [10]. Finally, linear SVM classifiers are used to predict the presence of an object within each region and to recognize object categories. Although RCNN outperforms older approaches, it has significant limitations. The use of redundant feature computations on a large number of overlapped suggestions (around 2000 boxes from a single image) results in an exceedingly poor detection speed. The selective search algorithm is a fixed algorithm as well. As a result, no learning occurs at that stage. This may result in the creation of poor candidate region ideas.

b) SPPNet: K. He et al. presented Spatial Pyramid Pooling Networks in 2014 [11]. Traditionally, there is a single pooling layer or no pooling layer at the transition between the convolution layer and the fully connected layer. SPPNet recommends having numerous pooling layers with varying scales. Previous CNN models, too, require a fixed-size input. SPPNet's Spatial Pyramid Pooling (SPP) layer allows a CNN to provide a fixed-length representation regardless of the size of image/region of interest without rescaling it.

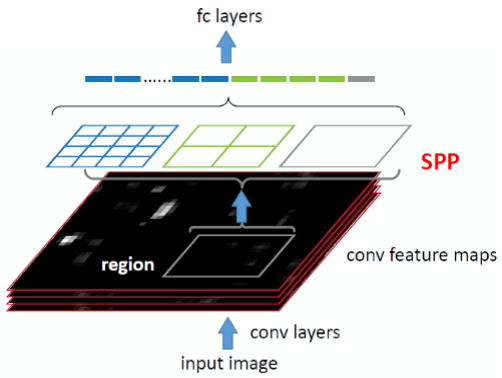

Figure 2: SPPNet

The figure above illustrates the process. We see that the input image goes to SPPNet using convolution network only once. Selective Search is used to generate region proposals just like in R-CNN. At the last convolution layer, feature maps bounded by each region proposal is going into SPP layer then FC layer [11].

SPPNet processes the image at conv layers only once, whereas R-CNN processes the image at conv layers as many times as there are region proposals. However, there are some drawbacks: training is still multi-stage, and SPPNet only fine-tunes its fully connected layers while ignoring all preceding layers.

c) Fast RCNN: In 2015, R. Girshick proposed Fast RCNN detector [12], which is a further improvement of R-CNN [10] and SPPNet [11]. Compared to an R-CNN model, a Fast R-CNN model uses the entire image as the CNN input for feature extraction, rather than each proposed region. Selective search is applied on the image and suppose it generates n proposed regions, their different shapes indicate regions of interests (RoIs) of different shapes. Fast R-CNN introduces RoI pooling, which uses the CNN output and RoIs as input to output a concatenation of the features extracted from each proposed region and fed into a fully connected layer During category prediction, the shape of the fully connected layer output is again transformed to n×q and we use softmax regression (q is the number of categories and n is the number of proposed regions). During bounding box prediction, the shape of the fully connected layer output is again transformed to n×4. This means that we predict the category and bounding box for each proposed region.

The reason "Fast R-CNN" is faster than R-CNN is because we don't have to feed all region proposals to the convolutional neural network every time. Instead, the convolution operation is done only once per image and a feature map is generated from it [12]. Although Fast-RCNN successfully integrates the advantages of R-CNN and SPPNet, its detection speed is still limited by the proposal detection.

d) Faster RCNN: In 2015, S. Ren et al. proposed Faster RCNN [13] detector shortly after the Fast RCNN. It is the first end-to-end, and the first near-real time deep learning object detector. All of the above algorithms (R-CNN, SPPNet & Fast R-CNN) uses selective search to find out the region proposals. Selective search is a slow and time-consuming process affecting the performance of the network. Faster R-CNN eliminates the selective search algorithm and lets the network learn the region proposals.

Although Faster RCNN breaks through the speed bottleneck of Fast RCNN, there is still computation redundancy at subsequent detection stage. Later, a variety of improvements have been proposed, including RFCN and Light head RCNN.

e) Feature Pyramid Networks (FPN): T.-Y. Lin et al. proposed Feature Pyramid Networks in 2017 [14]. When we look more at Faster RCNN, we notice that it is mainly incapable of detecting small objects in the image. To address this, a simple picture pyramid can be used to scale images over various sizes before sending them to the network. Once the detections on each scale have been identified, all of the predictions can be integrated using various approaches.

Before FPN, most of the deep learning-based detectors run detection only on a network's top layer. Although the features in deeper layers of a CNN are beneficial for category recognition, it is not conducive to localizing objects [14]. A top-down architecture with lateral connections is developed in FPN for building high-level semantics at all scales. Since a CNN naturally forms a feature pyramid through its forward propagation, the FPN shows great advances for detecting objects with a wide variety of scales.

2.2.2.2 CNN based One-Stage Detectors:

a) You Only Look Once (YOLO): YOLO was proposed by R. Joseph et al. in 2015 [15]. All of the previous object detection algorithms use regions to localize the object within the image. The network does not look at the complete image, instead it looks at parts of the image which have high probabilities of containing the object.

YOLO trains on full images and directly optimizes detection performance. With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. It also predicts all bounding boxes across all classes for an image simultaneously. It divides the input image into an S × S grid. If the centre of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes [15]. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box it predicted is.

In spite of its great improvement of detection speed, YOLO suffers from a drop of the localization accuracy compared with two-stage detectors, especially for some small objects. YOLO's subsequent versions (YOLO V2, YOLO V3 and the latest YOLO V4) has paid more attention to this problem.

b) Single Shot MultiBox Detector (SSD): SSD was proposed by W. Liu et al. in 2015. Then in November 2016 the paper about SSD: Single Shot MultiBox Detector was released by C. Szegedy et al. which reached new records in terms of performance and precision for object detection tasks [16]. It is a one-step object detector just like yolo. The main contribution of SSD is the introduction of the multi-reference and multi-resolution detection techniques, which significantly improves the detection accuracy of a one-stage detector, especially for some small objects.

To extract feature maps, it begins with the Base network. For high-quality picture classification, a typical pretrained network is utilized, which is shortened before any classification layers. C. Szegedy et al. employed the VGG16 network in their paper. Other networks, such as VGG19 and ResNet, can be employed and should yield satisfactory results. Following the base network, multi-scale feature layers, which are a sequence of convolution filters, are added. These layers gradually shrink in size to allow detection predictions at many sizes. Then, non-maximum suppression is employed to remove overlapping boxes and preserve only one box for each identified object.

c) RetinaNET: It is discovered that there is extreme class imbalance problem during training of dense one-stage detectors. And it is believed that this is the central cause, despite of its high speed and simplicity, why the performance of one-stage detectors is inferior to two-stage detectors. A new loss function named "focal loss" has been introduced in RetinaNet [17] in which lower loss is contributed by "easy" negative samples so that detector will put more focus on hard, misclassified examples during training. Focal Loss enables the one-stage detectors to achieve comparable accuracy of two-stage detectors while maintaining very high detection speed.

With ResNet + FPN as backbone for feature extraction, plus two task-specific subnetworks for classification and bounding box regression, RetinaNet achieves state-of-the-art performance and outperforms well-known two-stage detectors like Faster R-CNN [17].

2.3 Face Recognition

Face Recognition refers to a technology which identifies human face by measuring various facial features and matching it with the data present in the database of recorded faces, for authentication purposes. A traditional 2D face recognition system works with photos or videos gathered from surveillance systems, commercial/private cameras, CCTV, or other commonplace hardware.

We divide 2D face recognition approaches into four subclasses based on the nature of the extraction and classification methods used: (1) holistic methods, (2) local (geometrical) methods, (3) local texture descriptors-based methods, and (4) deep learning-based methods.

2.3.1 Holistic Methods: Holistic or subspace-based methods presume that any M-collection of facial photos contains redundancy that may be reduced by decomposing the tensor. These approaches provide a set of basis vectors reflecting a reduced spatial dimension (i.e., subspace) while retaining the original set of images. But these approaches are very pone to context changes and misalignments. Principal component analysis (PCA), known as eigenfaces [18], linear discriminative analysis (LDA), known as fisherfaces [19], and independent component analysis (ICA) [20] are the most common linear techniques employed for facial recognition systems. Abhishree et al. [21] (2015) suggested a method based on Gabor Filter (GFs) to extract features and enhance the performance of face recognition systems. GFs are employed for capturing aligned facial characteristics at specific angles.

2.3.2 Local (geometrical) Methods: Human face recognition relies heavily on attention and fixations. Attentive processes are typically guided by landmark qualities that are localized in the considered space using a salience map. The landmarks in the face are used to register facial features, the normalization of expressions and most commonly employed landmarks on the face are the tip of the nose, the tips of the eyes, the tips at the corners of the mouth, the eyebrows, the middle of the iris, the top of the ear, the nostrils, and nasal.

Pentland et al. [22] (1994) proposed a PCA version based on components in which the facial subspace was created from partial pictures of the initial facial photos. The landmarks chosen were caught between the mouth and the eyes. Tistarelli [23] (1995) suggested a system focused on extracting facial references re-sampled by a log-polar mapping program.

2.3.3 Local Texture Descriptors-based Methods: Feature extraction strategies focused on knowledge about the texture play a significant role in pattern recognition and computer vision. Ahonen et al. [24, 25] (2004–2006) presented a novel and effective representation of the facial image based on the local texture descriptor named: local binary pattern (LBP). The facial picture was divided into blocks, and the distributions of the LBP feature were chosen and integrated into a better histogram that was utilized as a facial descriptor. Then in 2006, Rodriguez and Marcel [26] (2006) suggested a generative method to face verification based on LBP facial representation as a complementary work. They created a universal facial model as a series of LBP-histograms; the histograms extracted from each block were seen as a distribution probability rather than a statistical observation. Kannala and Rahtu [27] (2012) proposed a method for building local image descriptors that encode

texture information efficiently and are proper for the description of image regions based on histograms.

2.3.4 Deep Learning based Methods: Deep learning can be categorized into three main classes depending on how the technique and architecture are used:

- Unsupervised or generative (Auto encoder (AE) [28], Boltzman Machine (BM) [29], Recurrent Neural Network (RNN) [30])
- Supervised or discriminative (CNN);
- Hybrid (Deep Neural Network (DNN) [31]).

i) CNN: CNNs consist of a set of filters/kernels/neurons with learnable parameters or weights and biases which have been added. Each filter takes some inputs, makes convolution, and follows it with a non-linearity [32]. The structure of CNN includes layers of convolutional, pooling, rectified linear unit, and fully connected. Some popular CNN architectures include LeNet, AlexNet, GoogleNet, VGGNet, ResNet, and SENet.

ii) Deep CNN based Methods: DeepFace, proposed by Taigman et al. [33] (2014), is a multi-stage technique that uses a generic 3D shape model to align faces. They created a facial representation using a 9-layer deep neural network trained on over 4000 identities in a multi-class face recognition task. One of the main challenges of face recognition is to develop an efficient feature representation for reducing intra-personal variations while increasing inter-personal variations, which can be solved with the Deep IDentification verification features (DeepID2) [34]. Liu et al. [35] (2016) have proposed a generalized Large-Margin Softmax Loss (L-Softmax), which combines the most generally used components in deep CNN architectures, that are: a cross-entropy loss, a Softmax loss, and the final fully connected layer. Zheng et al. [36] (2018) introduced a feature normalization approach for deep CNN, called ring loss, to normalize all samples of facial features via convex augmentation of the standard loss function (like Softmax). Ring loss applies soft feature normalization, where it ultimately learns to constrain facial feature vectors on the unit hypersphere.

3. METHODOLOGY

In this project, we have implemented IVS with three major schemes which are motion detection, face recognition, object detection.

3.1 Motion Detection

The background subtraction technique which is based on a static background hypothesis was used. Moving objects are detected by calculating the difference between the current frame and the base frame which serves as a reference frame. The frame is first converted to gray scale and then Gaussian Blur is applied to the gray image by passing it through a Low Pass Filter to smoothen it and remove possible noise in the image. The difference between the current frame and the base frame is then calculated.

$$Delta_{frame} = \left| Frame(x) - Base(x) \right|$$

$$for\ every\ pixel\ x\ in\ the\ frame$$

Image binarization is then applied to the Delta frame. This makes all pixels greater than or equal to the set threshold value be converted to white and all other pixels are converted to black pixels. This produces a binary image with only pure white or pure black pixels.

$$Thresh_{frame} = \begin{cases} 255, & \text{if x} \geq \text{threshold} \\ 0, & otherwise \end{cases}$$

The binarization of the frame make it possible to find the contours which is a curve drawn along the perimeter or boundary having the same color and intensity. Contours with very small area are filtered out as they may be due to noise as a result of wind or light fluctuations. The resulting contours are bounded by rectangles and the bounded rectangles are imposed on the frame which is then displayed.

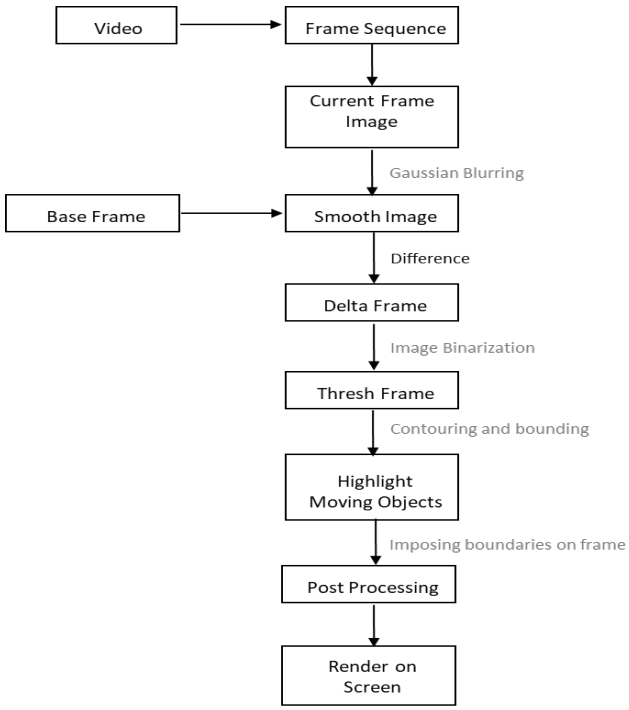

Figure 3: Block diagram of the Background Subtraction motion detection model

Figure 4: Output after the different stages of the Background Subtraction

3.2 Face Recognition

The face recognition involves 2 steps face detection, and then classification. The face detection part entails the detection of a face in the frame. The classification part is the conventional supervised machine learning problem which aims to label a detected face with the proper identity using the extracted features.

3.2.1 Face Detection

A) Face Detection with Haarcascades

This method was first proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001 [37]. It is a machine learning method of detecting faces where a cascade function is trained from a lot of positive and negative images (images with faces and images without faces) and is then used to detect faces in other images. The facial features are then extracted using Haar features which operate similar to a cross-correlation kernel [37]. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.

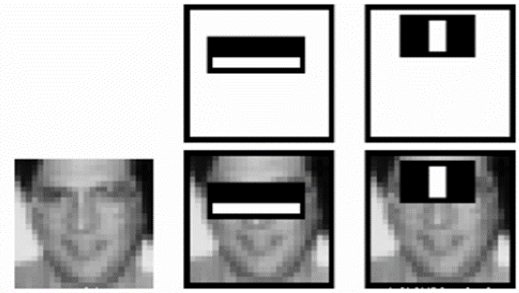

Figure 5: Applying the Haar features on the image

B) Face Detection with MTCNN

The Multi-Task Cascaded Convolutional Neural Network was first proposed by Kaipeng Zhang, et al. in the 2016 paper titled "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks" [38]. This network has a unique structure that cascades three network structures. The image is first rescaled into hierarchical sizes to form the image pyramid. The first model Proposal Network (P-Net) proposes candidate facial regions using a shallow CNN then the second model Refine Network (R-Net) filters the bounding boxes using a more complex CNN and the final model Output network (O-net) proposes the facial landmarks.

The three sub-models are trained on three different tasks which are face classification, bounding box regression and landmark localization and outputs of previous models are fed as inputs to subsequent models thus forming a three-network chain.

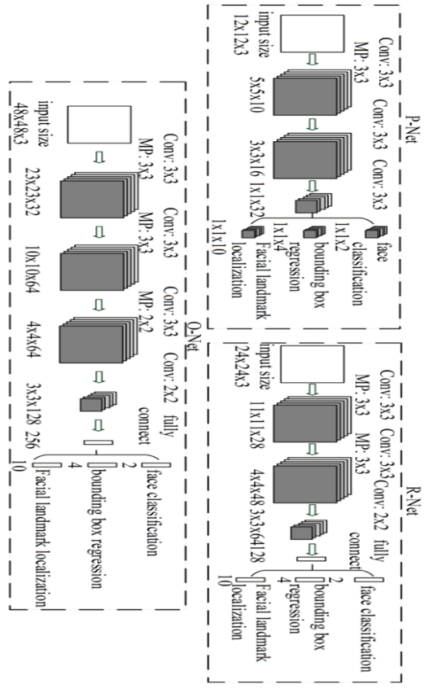

Figure 6: Model diagram of the MTCNN network

3.2.2 Classification

After the facial features have been extracted in a vectorial form, the embeddings are fed as inputs to classification models for the actual labelling of the face. In this work, three different classification models were implemented, and their performances were compared against each other.

3.2.2.1 Logistic Regression

This is a popular machine learning classification method that uses a sigmoid function to bound the output of a linear equation to a non-linear curve in the range of 0 and 1.

$$\text{sigmoid}(z) = \frac{1}{1 + exp(-z)}$$

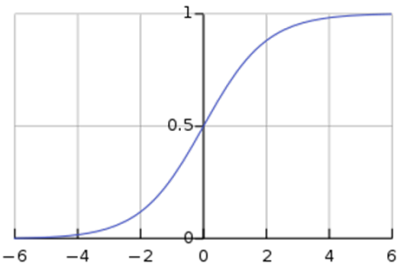

Figure 7: Graph of the sigmoid function

The probability of any observation belonging to a particular class is given by the equation –

$$P(y^{(i)} = 1) = \frac{1}{1 + exp(-(\beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_p x_p^{(i)}))}$$

$where\ \beta_0,\ \beta_1,\ \ldots,\beta_p$ represent the weights

The logistic regression model is trained to update the weight parameters which is associated with each value in the feature matrix. The optimum weight solution is found using gradient descent by taking steps proportional to the negative of gradient towards an optimum solution. A cost function is also used to measure how far a particular solution is from the optimal solution. The cost function and gradient descent calculation is repeated multiple times and the weights are updated with each iteration.

3.2.2.2 Kernel Support Vector Machine

SVM is yet another popular machine learning classification technique. It aims to separate the different target classes using a (n-1)-dimensional hyperplane in n-dimensional or multidimensional space. The hyperplane is a decision boundary and the model tries to maximize the margin.

The main motive of the SVM is to create the best decision boundary that can separate two or more classes (with maximum margin) so that we can correctly put new data points in the correct class. This process is quite direct if the data is linearly separable but for non-linearly separable data, we use the kernel trick. The kernel is a function that is applied to the data to move it to a higher dimension where it can be linearly separable. The kernel used here was the Gaussian Radial Basis Function due to its inherent ability to overcome the space complexity problem. Its equation is given as:

$$K(x,\ x') = \exp\left(-\frac{\left|\left|x - x'\right|\right|^2}{2\sigma^2}\right)$$

Here, σ controls the influence of new features on the decision boundary. The higher the sigma value, the more influence the features will have on the decision boundary.

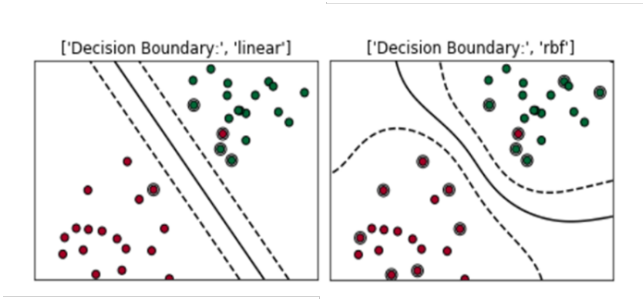

Figure 1: SVM classification of linear decision boundary and non-linear decision boundary using Gaussian RBF kernel.

3.2.2.3 Neural Network

The neural network performs a feature extraction operation on the detected face and then compares faces using the Euclidean distance between the extracted feature vectors. In the feature extraction, the neural network is used to extract facial feature from the detected face to output a vector which represents the facial embeddings. The facial embeddings represent important features of the detected face. The neural network is trained to output vectors which are similar for the same face and dissimilar for different faces. This means facial embeddings of the same face are close in the vector space and embeddings of different faces are far in the vector space.

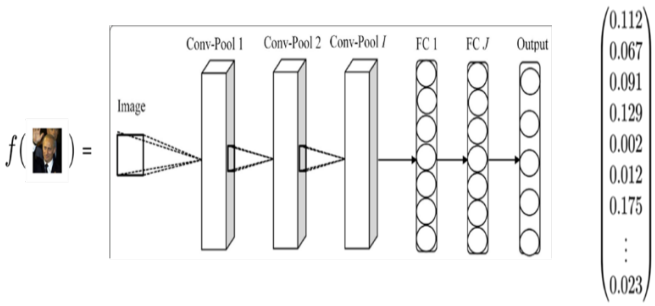

Figure 8: The network diagram of the neural network showing the feature extraction

A threshold of 0.6 was set so that if the distance between any two face embeddings is greater than the threshold then the model classifies them as different and if it is less, the model classifies the two faces as the same.

$$y = \begin{cases} 1, & if\ dist(x1, x2)\ \leq 0.6 \\ 0, & otherwise \end{cases}$$

3.3 Object Detection

Object detection was incorporated in this work to give the IVS the ability to detect objects in the environment. Three popular object detection algorithms were implemented and compared against each other.

3.3.1 ResNet

A residual neural network (ResNet) is an artificial neural network (ANN) that utilizes skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double or triple layer skips that contain nonlinearities (ReLU) and batch normalization in between. This approach allows the data to flow easily between the layers without hampering the learning ability of the deep learning model. The skip connections help the model to bypass any layer that affects the performance of the model negatively thus preventing accuracy saturation. The skip layer also prevents the vanishing or exploding gradient problem that could arise when training the deep network thus also increasing the training speed. Skipped layers are gradually restored when the network learns its feature space.

In this work, the ResNet50 architecture was implemented. This stricture has a total of 50 layers. The first layer of the network is a convolution layer with a kernel size of 7 x 7 and 64 different kernels all with a stride of size 2.

- Next layer is a max pooling layer with a stride size of 2.
- The next convolution process on the pooled image uses 64 kernels of size 1 x 1. Following this a 3 x 3 x 64 kernel convolution and at lastly, a 1 x 1 x 256 kernel

convolution. These three layers are repeated 3 times giving a total of 9 convolution layers in this step.

- The next step is another convolution step with 1 x 1 x 128 kernel. After that a kernel of 3 x 3 x 128 and at lastly a kernel of 1 x 1 x 512 this step was repeated 4 times making a total of 12 convolution layers in this step.
- The next convolution phase uses a kernel of 1 x 1 x 256 and two more kernels of 3 x 3 x 256 and 1 x 1 x 1024 and this is repeated 6 times making a total of convolution 18 layers.
- The last convolution phase uses a 1 x 1 x 512 kernel with two more kernels of 3 x 3 x 512 and 1 x 1 x 2048. The setup is repeated 3 times making a total of 9 convolution layers in this phase.
- Finally, the resulting image is average pooled and flattened. The classification is done with a fully connected layer containing 1000 nodes using the SoftMax function.

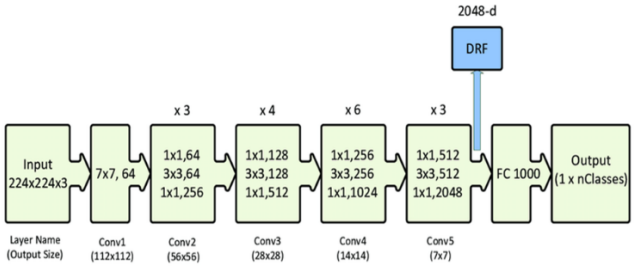

Figure 9: Model diagram of the ResNet50 Convolutional Neural Network

3.3.2 YOLO

This model was first described by Joseph Redmon, et al in the 2015 paper titled "You Only Look Once: Unified, Real-Time Object Detection" [39]. The YOLO model (You Only Look Once) is much faster than the ResNet in achieving object detection in real time. The model splits the input image into a grid where each cell in the grid is responsible for predicting a bounding box if the center of a bounding box falls within it [39]. Each grid cell predicts a bounding box and a confidence value. The bounding box is represented by a centroid (x, y coordinates), the width and height. The overall class prediction is based on the prediction of individual cells.

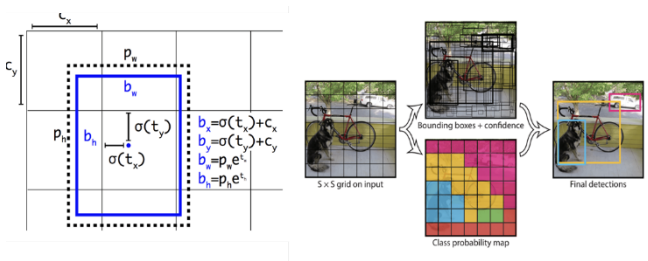

Figure 10: Bounding boxes with dimension priors and location prediction. The width and height are predicted as offsets from the cluster centroid.

The YOLOv3 architecture is based on the Darknet-53 network architecture. The YOLOv3 has a total of 53 convolution layers which are all followed by a batch normalization layer and Leaky ReLU activation. There are no pooling layers used in this architecture to prevent the loss of low-level featured which could arise during down sampling as a result of pooling.

Layer	Filters size	Repeat	Output size
Image			416 × 416
Conv	32 3 × 3/1	1	416 × 416
Conv	64 3 × 3/2	1	208 × 208
Conv Conv Residual	32 1 × 1/1 64 3 × 3/1	Conv Conv Residual × 1	208 × 208 208 × 208 208 × 208
Conv	128 3 × 3/2	1	104 × 104
Conv Conv Residual	64 1 × 1/1 128 3 × 3/1	Conv Conv Residual × 2	104 × 104 104 × 104 104 × 104
Conv	256 3 × 3/2	1	52 × 52
Conv Conv Residual	128 1 × 1/1 256 3 × 3/1	Conv Conv Residual × 8	52 × 52 52 × 52 52 × 52
Conv	512 3 × 3/2	1	26 × 26
Conv Conv Residual	256 1 × 1/1 512 3 × 3/1	Conv Conv Residual × 8	26 × 26 26 × 26 26 × 26
Conv	1024 3 × 3/2	1	13 × 13
Conv Conv Residual	512 1 × 1/1 1024 3 × 3/1	Conv Conv Residual × 4	13 × 13 13 × 13 13 × 13

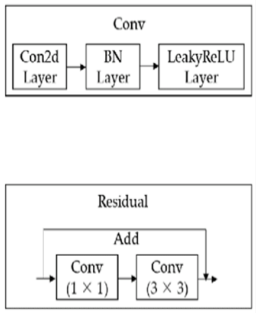

Figure 11: Network configuration of the YOLOv3 model

4. RESEARCH QUESTIONS AND HYPOTHESIS

1. How effective is the background subtraction technique of motion detection under varying environmental conditions?
2. Which method is more suitable for face detection between Haarcascades and MTCNN?
3. Which of the classification methods between CNN, Logistic Regression and Kernel SVM is most suitable for recognizing faces in application to IVS?
4. Which of the object detection models between ResNet and YOLO is more suitable for IVS application?

To answer the first research, question the background subtraction algorithm developed in this work will be tested with real time surveillance under varying environmental conditions. The environmental conditions that would be varied our lights and wind the performance of the glory team will be estimated by visual inspection.

For the second research question the two different face detection techniques will be tested with four different images with a total of 100 faces. The faces are well varied with some tilted some giving gestures some wearing glasses I'm some even slightly covered. The accuracy of the detection technique will be measured by the number of faces each method is able to detect from the datasets.

For the third research question the famous Labeled Faces in the Wild (LFW) dataset of face photographs will be used. This dataset was designed for studying the problem of unconstrained face recognition and it consists of 13,233 images and 5759 identities. This database was created and maintained by researchers at the University of Massachusetts, Amherst. In this experiment will be computing the accuracy of the different classification models under varying training sizes. Considering that the aim of this

work is for implementation of IVS in real time video surveillance we will also be comparing the time performance off the different methods.

For the last research question the Virat Video Dataset taken from CCTV security footages which includes actions performed by the general population at different locations will be used to compare the time and detection performance of the different object detection models. A set of random images collected over the internet will also be used to compare the detection performance of the three models.

5. RESULTS AND DISCUSSIONS

5.1 Efficiency of Background Subtraction Technique

The Background Subtraction technique proved to be a simple and computationally inexpensive technique but in terms of detection performance, it did not perform very well as it is highly susceptible to noise due to its assumption of a static reference background. In indoor test with uniform lighting, it performed really well in detecting motion in the scene but when there are lighting variations, it starts to detect many false positives as shown in figure 12. The noise effects are visible in the white contours in the thresh frame of figure 12 (b) and (c) and the false detections is shown in the processed frames.

In an outdoor environment, illumination changes and wind had an adverse effect on the performance of the Background subtraction technique as they introduce some noise into the frame. These environmental factors cannot be mitigated and a good motion detection technique should be robust and effective in the presence of these factors.

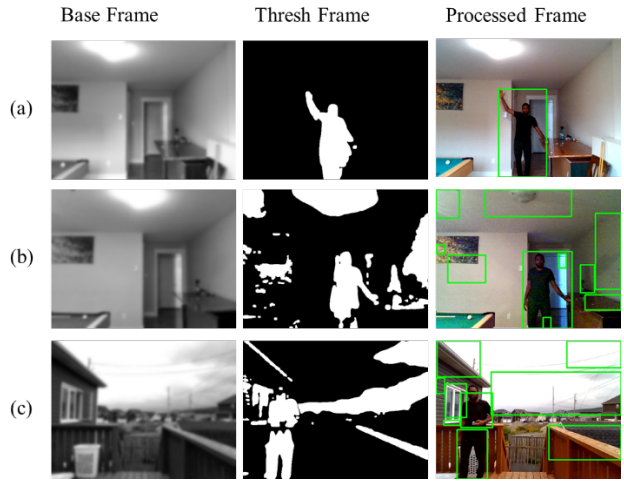

Figure 12: Effects of varying environmental conditions on the Background Subtraction Motion Detection. (a) shows the results with uniform lighting indoors. (b) shows the result with varying lighting indoors. (c) shows the result in an outdoor environment.

5.2 Haarcascades Vs MTCNN in face detection

The Haarcascades technique had a detection accuracy of 72 percent compared to the 87 percent detection accuracy of MTCNN. A closer inspection of some of the input images and the output of the detection algorithms showed that Haarcascades could not effectively identify distorted faces. Haarcascade detector could not identify faces with covered eyes or tilted heads effectively. MTCNN on the other hand showed a higher degree of freedom and could detect partially covered faces and tilted faces. From the sample images in figure 13, Haarcascades also seem to be more prone to false positives as it falsely detected some non-faces as faces. In terms of computation, the Haar classifier was more computationally efficient and took less execution time (average of 4.3 seconds for the image dataset) compared to MTCNN which had an average execution time of 8.1 seconds.

Figure 13: Comparison of the outputs of Haarcascade face detector (left column) and MTCNN face detector (right column)

5.3 Face Classification

To compare the three classification methods, experiments were carried out using the LFW dataset with varying training size. The training set was varied through 50%, 40%, 30%, 20%, 10%, and 5% of the dataset. The prediction results of the three models were computed and is shown in figure 14.

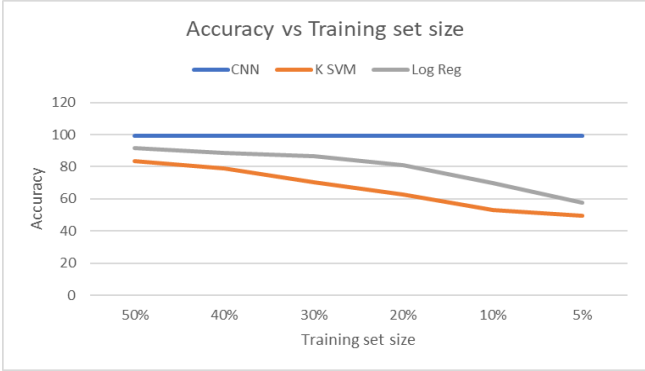

Figure 14: Accuracy of the CNN, Kernel SVM and Logistic regression classification models

From the plot, the accuracy of Kernel SVM and Logistic regression decreased as the training size decreased. The accuracy of the Deep network on the other hand remained the same. This can be attributed to the fact that it is able to extract important features from the face (face embeddings) which is then used to in calculating similarity or dissimilarity between faces.

The deep network using a feature extraction technique is able to recognize a face from just one sample by getting the face embedding and computing the distance between the new observation and the saved embedding and using a threshold value to classify the faces as the same or different. In terms of computation time, the training time and test time

for the different models was measured and is shown in table 1. The result shows that for its high accuracy, CNN has a computational trade off as it has a high execution time in testing. Logistic regression had the least testing execution time. In training, Kernel SVM had the best time efficiency.

Table 1: Table showing the computational time for training and testing the classification models

Model	**Average training time**	**Average testing time**
CNN	1420ms	49800ms
Kernel SVM	979ms	1200ms
Logistic Regression	1610ms	36.9ms

5.4 Object Detection

In the first experiment, the two models were tested on a set of random images collected over the internet. The results of the models on one of the samples can be seen in figure 15 and 16. ResNet50 and YOLOv3 had comparatively similar detection performance in figure 15 which is a traffic setting. In figure 16 which is a restaurant setting, YOLOv3 performed slightly better than ResNet as it could identify the dining table and donut. ResNet on the other hand identified the cell phone on the table which YOLOv3 did not. Overall, in terms of detection performance on the image database, both models perform comparatively equally but ResNet50 outperforms YOLOv3 in the detection of small objects.

Figure 15: Figure showing the detection output of ResNet50 and YOLOv3 on a test image (traffic)

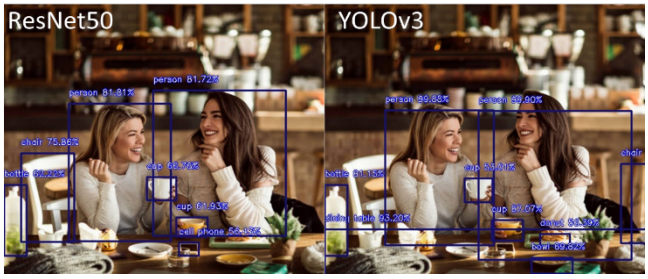

Figure 16: Figure showing the detection output of ResNet50 and YOLOv3 on a test image (restaurant)

The computation efficiency of the two models were tested using videos from the Virat Video Dataset and the metric used was the inference time which is the average time it takes the model to process an image. Yolov3 had an average inference time of 31ms while ResNet50 had an average inference time of 88ms. This mean that if applied to real time object detection, the maximum frame rate of the Resnet50 will be 11.4fps while that of YOLOv3 will be 32.3fps. The relatively low frame speed of Resnet50 make it less suitable for real time intelligent video surveillance.

6. CONCLUSION

The purpose of this work was to implement an effective Intelligent Video Surveillance system which incorporates motion detection, face recognition and object detection. From the experiments carried out, the Background Subtraction technique for motion detection is only suitable for indoor applications as its performance is highly susceptible to noise due to changing environmental factors like light intensity, shadows, and wind.

For the face detection aspect of the face recognition, Haarcascade detection was more time efficient but will be less suitable due to its poor detection accuracy on distorted or tilted faces. MTCNN proved to be a better face detection model as it is more robust to variations in facial orientations and distortions. For the classification part of the facial recognition, the neural network has high computational time but its ability to recognize faces from only one sample by creating a facial embedding of the face which highlights all the important facial features make it more applicable in Intelligent Video Surveillance because in real life situations, we may not have the luxury of getting multiple and varied pictures of a person to train the other models to properly identify the face of a person.

Finally, for the object detection, YOLOv3 is a better network model than the ResNet50 model. Although it struggles in the detection of very small objects unlike ResNet50, it still has comparable overall detection accuracy, and its inference time is less than 50% of the inference time of ResNet50. This inference time difference between the two models is very vital at it means that the frame speed of the YOLOv3 is about 32.3fps while that of ResNet50 is merely 11.4fps which will be too slow if used to implement real time video surveillance.

7. ACKNOWLEDGMENT

This work was done as a requirement for the completion of the ECE 659 course at the University of Waterloo.

REFERENCES

[1] R.T. Collins, A.J. Lipton, and T. Kanade, Introduction to the special section on video surveillance, IEEE Transactions on pattern analysis and machine intelligence 22(8) (2000), 745–746. doi:10.1109/TPAMI.2000.868676.

[2] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1. IEEE, 2001, pp. I–I.

[3] P. Viola and M. J. Jones, “Robust real-time face detection,” International journal of computer vision, vol. 57, no. 2, pp. 137–154, 2004.

[4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 886–893.

[5] D. G. Lowe, “Object recognition from local scale-invariant features,” in Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2. IEEE, 1999, pp. 1150–1157.

[6] ——, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004
[7] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, pp. 1–8.
[8] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, “Cascade object detection with deformable part models,” in Computer vision and pattern recognition (CVPR), 2010 IEEE conference on. IEEE, 2010, pp. 2241–2248.
[9] R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester, “Object detection with grammar models,” in Advances in Neural Information Processing Systems, 2011, pp. 442–450.
[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580– 587.
[11] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in European conference on computer vision. Springer, 2014, pp. 346–361.
[12] R. Girshick, “Fast R-CNN,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.
[14] T.-Y. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, ´ and S. J. Belongie, “Feature pyramid networks for object detection.” in CVPR, vol. 1, no. 2, 2017, p. 4.
[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.
[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, ´ “Focal loss for dense object detection,” IEEE transactions on pattern analysis and machine intelligence, 2018.
[18] Turk, M.; Pentland, A. Eigenfaces for recognition. J. Cogn. Neurosci. 1991, 3, 71–86.
[19] Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J. Eigenfaces vs Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 1997, 19, 711–720.
[20] Stone, J.V. Independent component analysis: An introduction. Trends Cogn. Sci. 2002, 6, 59–64.
[21] Abhishree, T.M.; Latha, J.; Manikantan, K.; Ramachandran, S. Face recognition using Gabor Filter based feature extraction with anisotropic diffusion as a pre-processing technique. Procedia Comput. Sci. 2015, 45, 312–321.
[22] Pentland, A.; Moghaddam, B.; Starner, T. View-Based and modular eigenspaces for face recognition. In Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 21–23 June 1994; pp. 84–91.
[23] Tistarelli, M. Active/space-variant object recognition. Image Vis. Comput. 1995, 13, 215–226.
[24] Ahonen, T.; Hadid, A.; Pietikäinen, M. Face recognition with local binary patterns. In Proceedings of the 8th European Conference on Computer Vision (ECCV), Prague, Czech Republic, 11–14 May 2004; pp. 469–481.
[25] Ahonen, T.; Hadid, A.; Pietikäinen, M. Face description with local binary patterns: Application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2006, 28, 2037–2041.
[26] Rodriguez, Y.; Marcel, S. Face authentication using adapted local binary pattern histograms. In Proceedings of the 9th European Conference on Computer Vision (ECCV), Graz, Austria, 7–13 May 2006; pp. 321–332.
[27] Kannala, J.; Rahtu, E. BSIF: Binarized statistical image features. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR), Tsukuba, Japan, 11–15 November 2012; pp. 1363–1366.
[28] Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 2010, 11, 3371–3408.
[29] Salakhutdinov, R.; Hinton, G. Deep Boltzmann machines. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater, FL, USA, 16–19 April 2009; pp. 448–455.
[30] Sutskever, I.; Martens, J.; Hinton, G. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue, WA, USA, 28 June–2 July 2011; pp. 1017–1024.
[31] Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Inf. Process. 2014, 3, 1–29.
[32] Guo, G.; Zhang, N. A survey on deep learning-based face recognition. Comput. Vis. Image Underst. 2019, 189, 10285.
[33] Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
[34] Sun, Y.; Chen, Y.; Wang, X.; Tang, X. Deep learning face representation by joint identification-verification. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 1988–1996.
[35] Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large margin softmax loss for convolutional neural networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 507–516.
[36] Zheng, Y.; Pal, D.K.; Savvides, M. Ring Loss: Convex Feature Normalization for Face Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5089–5097.
[37] Viola, Paul A., and Michael J. Jones. “Rapid object detection using a boosted cascade of simple features.” Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 1 (2001): I-I.
[38] Zhang, Kaipeng, Zhanpeng Zhang, Zhifeng Li and Y. Qiao. “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks.” IEEE Signal Processing Letters 23 (2016): 1499-1503.
[39] Chaudhari, Sujata, Nisha Malkan, Ayesha A. Momin and M. Bonde. “Yolo Real Time Object Detection.” International Journal of Computer Trends and Technology 68 (2020): 70-76.