**CHAPTER II**

**REVIEW OF RELATED LITERATURE**

A survey of related studies was undertaken by the researchers to get an insight from comparable studies to get suggestion regarding the ways and means for the collection of relevant data and interpretation of results.

**2.1 OpenCV**

OpenCV the open source computer vision library is released under a BSD license and hence it's free for both academic and commercial use. lt has a C++, C, Python and Java language support and supports Windows, Linux, Mac OS, iOS and Android operating systems. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can also take advantage of multi-core processing. [1]

**2.2.1 OpenCV Face Detection**

A recognition process can be much more efficient if it is based on the detection of features that encode some information about the class to be detected. This is the case of Haar-like features that encode the existence of oriented contrasts between regions in the image. A set of these features can be used to encode the contrasts exhibited by a human face and their spacial relationships. Haar-like features are so called because they are computed similar to the coefficients in Haar wavelet transforms. [1]

The object detector of OpenCV has been initially proposed by Paul Viola and improved by Rainer Lienhart. First, a classifier, namely a cascade of boosted classifiers working with haar-like features is trained with a few hundreds of sample views of a particular object, and negative examples which are arbitrary images of the same size. [41]

After a classifier is trained, it can be applied to a region of interest in an input image. The classifier outputs a "1" if the region is likely to show the object, and "0"othenrise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales. [41]

The process of cascading means that the resultant classifier consists of several simpler classifiers stages that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word "boosted" means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques called weighted voting. [1]

Currently Discrete AdaBoost, Real AdaBoost, Gentle AdaBoost and Logitboost are supported. The basic classifiers are decision-tree classifiers with at least two leaves. Haar-like features are the input to the basic classifiers. The feature used in a particular classifier is specified by its shape, position within the region of interest and the scale. [1]

**2.2.1.1 Rapid Object Detection using a Boosted Cascade of Simple Features**

A research papers presented on the Conference on Computer Vision and Pattern Recognition (2001) describes an approach for visual object detection capable of processing images rapidly along with high detection rates using Machine Learning. An integral image is introduced in their study which allows for the detector to compute features very quickly. A learning algorithm based on AdaBoost was also developed in their study which selects a small portion of critical visual features from a larger set which results as they called an extremely efficient set of classifiers. Then finally, they cascade this classifiers to get more detail and more complexity allowing background regions to be quickly discarded and the region of interests identified. [48]

**2.2.1.2 AdaBoost**

In machine learning, boosting is the approach of creating a highly accurate rules by combining the results of many weak and inaccurate rules. One of the first practical boosting algorithm was of Freund and Schapire which is mostly used and studied today. The algorithm assumes that each weak hypothesis has accuracy that is better than random guessing. This assumption is sometimes called a weak learning condition. Given the weak learning condition it is then possible to prove that the training error according to Schapire can be reduced to zero very rapidly.[49]

**2.2.1.3 FPGA-Based Face Detection System Using Haar Classifiers**

A comparable study presents a hardware architecture for face detection designed using the AdaBoost algorithm using Haar features. The proponents of this study designed an image scaling, integral image generation, pipelined processing as well as a classifier and parallel processing multiple classifiers to accelerate the processing speed of their face detection system. They designed their system using Verilog HDL and implemented it in Xilinx Virtex 5 FPGA. They were able to show 35 times increase in performance compared to equivalent software implementation.[50]

**2.2.1.4 Object Detection Using the Statistics of Parts**

A research study by Schnedierman and Kanade proposes an object detector and its instantiations for detecting faces and cards at any size, location and pose. Their implementation uses multiple classifiers from different orientations. Each of such classifiers determines the presence of an object on a specified image window. The classifiers scan the image exhaustively such that the location and size of the target can be determined. [51]

**2.2.1.5 A Parallel Architecture for Hardware Face Detection**

Theocharides and his group also studied a scalable parallel architecture for face detection using AdaBoost algorithm. Their experimental results show that their proposed architecture can detect faces at the same accuracy as the software implementation on a real-time video at 52 frames per second. [52]

**2.2.2 OpenCV Face Recognition**

Presently, OpenCV supports three different algorithms for Face Recognition namely, Eigenfaces; Fisherfaces; and Local Binary Patterns Histograms. Face recognition is an easy task for humans. lt was shown by David Hubel and Torsten Wiesel, that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since humans don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some kind of classification on them. [42]

In computerized face recognition, each face is represented by a large number of pixel values. Linear discriminant analysis is primarily used here to reduce the number of features to a more manageable number before classification. Each of the new dimensions is a linear combination of pixel values, which form a template. The linear combinations obtained using Fisher's linear discriminant are called Fisher faces, while those obtained using the related principal component analysis are called Eigen faces. [42]

**2.2 Current State of OpenCV Acceleration.**

There have been many efforts in accelerating the current OpenCV library. However, none of them are focused on the ARM architecture which is the de facto standard in mobile and embedded applications. It is of special interest for this research that the status for Hardware Acceleration of OpenCV to be studied due to the fact that this research will lead to improved chances of increasing the possibility of hardware acceleration of OpenCV on ARM.

**2.2.1 OpenCV GPU**

The OpenCV GPU module is a set of classes and functions to utilize GPU computational capabilities. lt is implemented using NVIDIA CUDA Runtime API and supports only NVIDIA GPUs. The OpenCV GPU module includes utility functions, low level vision primitives, and high-level algorithms. The utility functions and low-level primitives provide a powerful infrastructure for developing fast vision algorithms taking advantage of GPU whereas the high-level functionality includes some state-of-the-art algorithms (such as stereo correspondence, face and people detectors, and others) ready to be used by the application developers. [43]

**2.2.2 OpenCV IPP**

lntel lntegrated Performance Primitives (lntel IPP) is an extensive library of multicore-ready, highly optimized software functions for multimedia, data processing, and communications applications. lntel IPP offers thousands of optimized functions covering frequently used fundamental algorithms. There is a free non-commercial version of IPP for Linux as made available by lntel but the implementation is proprietary. [44] [1]

**2.2.3 OpenCV Applications with Zynq-7000 All Programmable SoC**

The design flow leverages HLS technology in the Vivado Design Suite, along with optimized synthesizable video libraries. The libraries can be used directly, or combined with application-specific code to build a customized accelerator for a particular application. This flow can enable many computer vision algorithms to be quickly implemented with both high performance and low power. The flow also enables a designer to target high data rate pixel processing tasks to the programmable logic, while lower data rate frame-based processing tasks remain on the ARM cores. [45]

Alternatively, the OpenCV function calls can be replaced by corresponding synthesizable functions from the Xilinx Vivado HLS video library. OpenCV function calls can then be used to access input and output images and to provide a golden reference implementation of a video processing algorithm. After synthesis, the processing block can be integrated into the Zynq Programmable Logic. Depending on the design implemented in the Programmable Logic, an integrated block may be able to process a video stream created by a processor, such as data read from a file, or a live real-time video stream from an external input. [45]

**2.3 USB Video Class**

This research also takes great care in making sure the available input imaging devices are supported by the Linux USB Video Class. The USB Device Class Definition for Video Devices, or USB Video Class, defines video streaming functionality on the Universal Serial Bus. Much like nearly all mass storage devices can be managed by a single driver because they conform to the USB Mass Storage specification, UVC compliant peripherals only need a generic driver. [46, 47]

The UVC specification covers webcams, digital camcorders, analog video converters, analog and digital television tuners, and still-image cameras that support video streaming for both video input and output. However, as stated on their website, due to the limited available man power and the broad scope of the UVC specification, the Linux UVC project will concentrate the development efforts on video input devices, especially webcams. In addition, video output devices are supported in bulk mode only and are therefore less favored. [46, 47]

It was noted that the Logitech C525 Webcam utilized by the researcher as an input image device is fully supported by the Linux UVC. It is therefore necessary that the appropriate Kernel Modules for the Linux Kernel be included in the system.