CHAPTER II

LITERATURE REVIEW

2.1 Literature Review

According to the research done by Agrawal & Samson, (2016) titled “A Review on Feature Extraction Techniques and General Approach for Face Recognition”, one of the many ways of achieving face recognition is to use what is called feature extraction where the algorithm looks directly at the image for features that are specific to human faces. The article goes into details on what the general approach for face recognition based on feature extraction works, and also the evaluation parameters used which is False Acceptance Rate (FAR) and False Rejection Rate (FRR). FAR is the probability that a system will incorrectly identify an individual, or fail to reject an imposter, while FRR is the probability that a system will fail to identify altogether (also known as Error Rate).

Based on a research article done by Ding & Tao (2016) titled “A Comprehensive Survey on Pose-Invariant Face Recognition”, there are two major types of face recognition capabilities: frontal face recognition, and pose-invariant face recognition. Frontal face recognition is the type that has been intensively studied and gradually matured in the past few decades thanks to the help of new technologies and techniques, and only concerns itself with frontal perspectives of faces. Pose-invariant face recognition according to Ding & Tao is the next crucial step to realizing the full potential of face recognition for real
world applications. The research article mentions three degrees of freedom of pose variation of the face: yaw, pitch, and roll; and discusses the existing methods that researchers have used to tackle this specific problem in the field of face recognition.

A research article done by Artiges, Caron, Ekenel, Grm, & Struc, (2017) titled “Strengths and Weaknesses of Deep Learning Models for Face Recognition Against Image Degradations” goes in depth regarding strengths and weaknesses of using Convolutional Neural Networks (CNN) in face recognition, specifically against low quality images. The article mainly discusses the ways an image can be classified and made low quality, and passes those images to three pre-trained CNN models; the models are VGG-Face, GoogLeNet, and SqueezeNet. The parameters used to lower the quality of the images include: blur, contrast/brightness, partial obstruction of the face, and noise. The research article concludes that the most challenging aspect to tackle in low quality images is blur. However, given proper architecture choices and training procedures for the CNN models, a deep learning model can be trained and made to detect faces in low quality images.

Based on research done by Hart, Prikner, & Hartova (2018) titled “Influence of Face Lighting on The Reliability of Biometric Facial Readers” which goes into detail the effects lighting has on faces, mainly the casting of shadows and how they affect face recognition accuracy, and reliability specifically on biometric facial readers used commercially. The research is done by increasing and decreasing luminosity of a fixed light source around the detector, detecting faces that are fixed in a position. The conclusion from the research is that lighting...
greatly affect these devices and that lighting is an important factor in face recognition.

Lastly, based on a journal article written by Boyko, Basytiuk, & Shakhovska (2018) titled “Performance Evaluation and Comparison of Software for Face Recognition based on Dlib and Opencv Library” where the researchers dive into one of the most popular problems in computer vision which is face recognition using the open-source library Dlib and Opencv library. The article only attempts to use Histogram of Oriented Gradients (HoG) in detail when doing their comparison and evaluation, and only compared the time needed to do face detection for a small number of images.
### Table 1: Literature review conclusion, and remarks

<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Title</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boyko, Basytiuk, &amp; Shakhovska</td>
<td>2018</td>
<td>Performance Evaluation and Comparison of Software for Face Recognition based on Dlib and Opencv Library</td>
<td>The research article evaluates mostly HoG in both libraries and compares the speed and accuracy against frontal face images on a small dataset.</td>
</tr>
<tr>
<td>Hart, Prikner &amp; Hartova</td>
<td>2018</td>
<td>Influence of Face Lighting on The Reliability of Biometric Facial Readers</td>
<td>The research article discusses and analyzes the effects various lighting types has on face detection (feature based).</td>
</tr>
<tr>
<td>Artiges, Caron, Ekenel, Grm &amp; Struc</td>
<td>2017</td>
<td>Strengths and Weaknesses of Deep Learning Models for Face Recognition Against Image Degradations</td>
<td>The research article discusses DNN and CNN based face recognition and how they perform against low quality images.</td>
</tr>
<tr>
<td>Agrawal &amp; Samson</td>
<td>2016</td>
<td>A Review on feature extraction techniques and general approach for face recognition</td>
<td>The research article focuses on feature based recognition, and discusses the different features that the algorithms look for. The research article uses False Acceptance Rate (FAR) and False Rejection Rate (FRR) as evaluation parameters.</td>
</tr>
<tr>
<td>Ding &amp; Tao</td>
<td>2016</td>
<td>A Comprehensive Survey on Pose Invariant Face Recognition</td>
<td>The survey only goes in depth on pose invariant face detection, and surveys the algorithms based on pose invariant.</td>
</tr>
</tbody>
</table>
In conclusion, based on the work already done by the authors mentioned above; There has been numerous research and studies done on various types of face recognition algorithms, with each of them looking at different aspects that could possibly affect the outcome of the detection. In this mini thesis; the writers will incorporate elements from all five studies into it, mainly the parameters used in each study.

2.2 Theoretical Base

2.2.1 Face Recognition

Out of the many things that a human being can do reliably well when compared to computers is the ability to be able to recognize faces in a reliable fashion. Face recognition systems generally work by analyzing patterns detected in the facial contours of the human face. With the help of big data, and the Internet of Things (IoT); face recognition systems has seen major improvements in terms of availability, and viability (Taigman, Yang, & Ranzato, 2014).

According to S. Agrawal & Khatri (2015), there are three major approaches when it comes to facial recognition in the field of computer vision:

1. Feature based approach

   In general, feature extraction is the kind of process where the input is reduced to its basic form, leaving only the important features behind. In feature based approaches, the facial features are detected and segmented before being put in as input into a classifier. This particular approach is useful in case of large amounts of images.
2. Holistic approach

In Holistic approaches, the whole face in its entirety is analyzed using statistical methods to extract the statistical characterization from the entire training sample of images.

Figure 1 outline of the features used for feature based face recognition, source: “Facilitating Fashion Camouflage Art” (2013).

Figure 2 Images of generated Eigen Faces, source: AT&T Laboratories
3. **Hybrid approach**

A Hybrid approach is basically where methods from both the feature, and holistic based approaches are combined. The idea comes from how humans perceive the human face both as a whole, and also the local features.

According research done by Shyam & Singh (2015) there are three techniques to face recognition in computing:

1. **Appearance based statistical techniques**

In recent years, large amounts of appearance based statistical techniques are proposed that are used to do face detection and recognition. These techniques look at hundreds to thousands of face images; and turn them into a feature map that is usable which will then be used for face recognition purposes. One technique that uses this is called Principal Component Analysis (PCA).

![PCA for facial recognition, source: 4.bp.blogspot.com/-Gk3UWoJe5oA/Uqqa_EJgszI/AAAAAAAAAxY/HQV9gZpyGuo/s640/fig1.jpg](https://4.bp.blogspot.com/-Gk3UWoJe5oA/Uqqa_EJgszI/AAAAAAAAAxY/HQV9gZpyGuo/s640/fig1.jpg)

**Figure 3** PCA for facial recognition, source: 4.bp.blogspot.com/-Gk3UWoJe5oA/Uqqa_EJgszI/AAAAAAAAAxY/HQV9gZpyGuo/s640/fig1.jpg
2. Texture based techniques

Texture based techniques work by taking a look at each of the pixel values of an image directly to determine whether it contains a face or not.

A popular method that uses this technique is called Local Binary Pattern (LBP) which has already seen usage in face recognition systems.

![Figure 4 LBP histogram feature extraction, source: www.advancedsourcecode.com/images/lbp_facerecognition.jpg](www.advancedsourcecode.com/images/lbp_facerecognition.jpg)

3. Multimodal techniques

A face recognition technique where the information gathered from multiple sources are combined together. By combining the various information gathered from different sources using an effective fusion scheme, we can significantly increase the model accuracy of a biometric face recognition system.

Large datasets are used by face recognition systems that depend on machine learning algorithms to be accurate; the general consensus that has been agreed upon is that the larger the dataset is, the better the algorithm will perform (Kemelmacher-Shlizerman, Seitz, Miller, & Brossard, 2015). This is the reason...
why datasets used by giant corporations like Google, and Facebook are private property of the company, and is not open for the public to see (Parkhi, Vedaldi, & Zisserman, 2015). However, even though that is the case; open source face recognition algorithms have propped up where the accuracy is comparable to those of Google’s and Facebook’s even though they are using smaller datasets.

With the help of public open source libraries like OpenCV, and Dlib; Face recognition technology has never been more accessible to the public; enabling individuals to create, and implement their own face recognition systems.

2.2.2 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a computing system loosely inspired from the brains of organic beings (mostly human) to process data and information. The structure of an ANN includes neurons similar to those found in actual neural networks of the brain to process input data (Hemmat Esfe, Saedodin, Sina, Afrand, & Rostami, 2015).

According to Jordan & Mitchell (2015) there are three approaches, also referred to as learning paradigms that are used to train, and prepare Artificial Neural Networks:

1. Supervised learning

Supervised learning is where the ANN is provided with the input-output pairs from existing data that has been labelled before. The ANN is then used to analyze the provided training data, and produce an inferred function. The training would result in the ANN correctly determining the correct class label from data it has not seen before.
2. Unsupervised learning

Unsupervised learning is where the ANN is not provided any context whatsoever; and is then expected to provide a satisfactory output. One area where Unsupervised learning is used is to do clustering to group together data based on similarities that they have.

3. Reinforced learning

Reinforced learning is where the ANN learns based only on the notion of a cumulative reward or score that it gets when it does something right, and reductions in said score when it doesn’t in some instances.

Aside from the learning paradigms mentioned above, there are also more than one type of ANNs architectures out there. In this research we will only be talking about Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) since the face recognition algorithms we will be analyzing are using both neural network architectures.

1. Deep Neural Network (DNN)

DNN is an ANN with a multilayer networks of threshold units hence the name “deep” networks, each of which computes some simple parameterized function of its inputs (Jordan & Mitchell, 2015). A complex DNN have many layers. DNNs have dramatically improved speech recognition, computer vision, and also other fields such as genomics (Lecun, Bengio, & Hinton, 2015).
2. Convolutional Neural Network (CNN)
According to Ashqar & Abu-naser (2019) CNN is a class of artificial neural network, most commonly applied to analyzing visual imagery that use a variation of multilayer perceptrons designed to require minimal preprocessing (convolutional layers). A CNN consists of multiple successively increasingly refined data filters which are the layers of the CNN, the input data to a CNN can be represented as a 3D tensor of size: height x width x channels of pixel values ranging from 0 to 255 (Ewald, Goby, Jansen, Groves, & Benedictus, 2018). Which is why CNNs are usually used in the field of computer vision.

2.2.3 Haar Cascade
Haar cascade (Haar-like features) is a classifier that uses features consisting of adjacent rectangular windows at specific locations (Pandya, 2016). The main advantage to using Haar-like characteristics is their feature to compile information of the described areas under certain conditions (Katek, Holik, Zablocki, & Dobrzynska, 2016).

2.2.4 Histograms of Oriented Gradients (HoG)
Histogram of Oriented Gradients (HoG) are feature descriptors used in computer vision and image processing for the purpose of object detection (Desai, Jadhavm, & Guravvagol, 2017). According to Patel, Thakar, & Shah (2016) HoG is used for human detection, the essential though behind HoG is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions.
2.3 Software Used

2.3.1. Sublime Text

Sublime Text is a text, and source code editor that supports multiple programming, and markup languages, complete with a syntax highlighter (Kulvanit, Jones, Bosworth, & Wetzel, 2015). In Sublime Text there is a package manager where plugins can be installed by the user, the packages are typically community-built and maintained under free-software licenses. The first released version of Sublime Text is developed by Jon Skinner, and Will Bond. Sublime Text was released on January the 18th, 2008 (Sumangali, Borra, & Suraj Mishra, 2017). Other than to serve as a text editor, one of Sublime Text’s feature where you can install plugins means the writers can install Python plugins and manage their python libraries with ease, this is especially helpful since the mini-thesis requires the use of multiple Python libraries.
2.3.2. Python 3.6.5

Python is a high-level, and an interpreted programming language for general-purpose programming, well known for its expressive, concise, and easy to read syntax; created by Guido van Rossum and first released in 1991 (Helmus & Collis 2016). The programming language has seen a large surge in popularity and usage due to its many positives; such as: readability, modularity, and large standard library (Muller et al., 2015).

2.4 Library Used

2.4.1 OpenCV

OpenCV is an open source computer vision library that is also free for commercial and non-commercial use (under the BSD license) (Palekar, Parab, Parikh, & Kamble, 2017). It has Python, Java, and C++ interfaces and is supported on multiple platforms. OpenCV was designed with computational efficiency in mind and with a strong focus on real-time applications in the field of computer vision (Suryatili & V.B.Dharmadhikari, 2015). It is estimated that OpenCV has more than 47 thousand people in its community, with over 14 million downloads. OpenCV can be used for a wide range of things; from interactive art, to advanced robotics (Sathyabama, 2017). From the OpenCV library we will use the DNN and HAAR Cascades algorithm.

The DNN model used is pre-trained already and are downloadable via the library’s GitHub repository. Same goes for the XML rules/templates for HAAR Cascades.
2.4.2 Dlib

Dlib is a general purpose open source software library that contains various algorithms that can be used for machine learning as well as tools for creating complex software to solve real world problems. Dlib is also supported on multiple platforms (Ucar & Hsieh, 2018). The library is commonly used in both academia and industry in a wide range of fields including (but not limited to) robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib’s open source licensing allows you to use it in any application, free of charge. From the Dlib library we’re using the CNN and HoG implementation of their face recognition algorithms. The CNN models used for Dlib is pre-trained and available for download from the library’s GitHub repository, as for the HoG algorithm; it’s already built into the library.

2.4.3 NumPy

NumPy is the fundamental Python library for scientific and mathematical computing (Pawlik, Segal, Sharp, & Petre, 2014). It contains among other things:

1. a powerful N-dimensional array object.
2. sophisticated (broadcasting) functions.
3. tools for integrating C/C++ and Fortran code.
4. useful linear algebra, Fourier transform, and random number capabilities.

Other than the library’s obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined in NumPy, this allows it to be seamlessly and quickly integrated into a wide variety of use cases (Pasumarti & Sekhar, 2018). NumPy is licensed under the BSD license, allowing reuse with minimal restrictions.