



ABDULLAH ASLAM

01-134172-005

AHSAN GOHEER

01-134172-008

On the Fly Deep Learning Based Video Search Engine

Bachelor of Science in Computer Science

Supervisor: Ghulam Ali Mirza

Department of Computer Science
Bahria University, Islamabad

May 2021

Abstract

Retrieval of information based on the content is a complicated process and requires a combination of different techniques to produce accurate results. The increase in computational power and the ability to process large amounts of data has enabled the use of deep learning algorithms to provide solutions to difficult problems. Nowadays deep learning algorithms are being used in content-based retrieval systems but there are currently no systems that allow real-time learning. Most deep learning based content retrieval systems are limited by their inability to adapt to new classes of data that were not originally a part of the model training process. This can be a hindrance when developing a system for retrieving specific person(s) from the data. It is not possible to train a model with a class for each individual person. This project aims to provide a solution to this problem by using a relatively new concept of deep learning known as “On the fly deep learning”.

In this approach the system is divided into two parts: Online processing and Offline processing. The offline processing deals with preparing the data, detecting the required person(s), extracting feature vectors, and storing the feature vectors. Similarly, the online processing involves fetching positive examples at run-time from the internet based on the user query, detecting the person(s) in the examples, extracting feature vectors, and then comparing the features with stored features using a ranking algorithm. This approach yields relatively fast and accurate results.

The approach is motivated by human cognition and is similar to how humans learn to recognize people. Humans make connection between the features of a person’s face and certain attributes that distinguish them from others. For example: If a face exists in the human memory and a person wants to recognize someone based on that face, they would compare the attributes of both faces and discern their similarity.

Contents

Abstract	i
1 Introduction	1
1.1 Project Introduction	1
1.2 Problem Description	2
1.3 Project Objectives	2
1.4 Project Scope	3
2 Literature Review	5
2.1 Background	5
2.2 Related Work	5
3 Requirement Specifications	9
3.1 Existing System	9
3.2 Proposed System	9
3.3 System Requirements	10
3.3.1 Functional Requirements	10
3.3.2 Non-Functional Requirements	11
3.4 Use Cases	12
3.4.1 Use Case Description	13
3.4.2 Use Case Diagrams	19
4 System Design	23
4.1 System Architecture	23
4.1.1 Presentation Layer	23
4.1.2 Business Logic Layer	24
4.1.3 Data Layer	24

4.2	Design Constraints	24
4.2.1	Data Format	24
4.2.2	Multi-core System	24
4.2.3	Adequate System Memory	24
4.2.4	Competent GPU/CPU	24
4.2.5	Operating System	25
4.3	Design Methodology	25
4.3.1	Increment:1	25
4.3.2	Increment:2	25
4.3.3	Increment:3	25
4.4	Use Case Diagram	27
4.5	Sequence Diagrams	28
4.5.1	Adding Video(s) to the Database	28
4.5.2	Deleting Video(s) from the Database	28
4.5.3	Retrieving Frame Details from the Database	29
4.5.4	Display List of All Indexed Persons	29
4.5.5	Update Video Database	30
4.5.6	Executing a Search Query	31
4.5.7	Administrator Login Sequence	31
4.6	Component Diagram	33
4.7	Deployment Diagram	34
4.8	Activity Diagram	35
4.9	High Level Diagram	36
4.10	System Flow Diagram	37
4.11	Data Flow Diagram	38
4.12	Database Diagram	39
5	System Implementation	41
5.1	System Architecture	41
5.1.1	Offline Processing	43
5.1.2	Online Processing	48
5.2	Datasets for training models	50
5.2.1	WIDER Face Dataset	50

5.2.2	VGG Face Dataset	52
5.3	Processed Video Dataset	54
5.4	Tools and Technology Used	55
5.4.1	Microsoft Visual Studio Code	55
5.4.2	GitHub	55
5.5	Development/Environment Language Used	56
5.5.1	Python	56
5.5.2	React Javascript	56
5.5.3	Material UI	56
5.5.4	Javascript	56
5.5.5	Babel	56
5.5.6	Web Pack	57
5.5.7	Django	57
5.5.8	Django Rest Framework	57
5.5.9	Node Package Manager	57
5.6	Implementation Strategy	57
6	System Testing and Evaluation	59
6.1	Graphical User Interface Testing	59
6.2	Usability Testing	59
6.3	Software Performance Testing	60
6.4	Compatibility Testing	60
6.5	Exception Handling	61
6.6	Security Testing	62
6.7	Comparative Analysis Between Host Machines	62
6.8	System Test Cases	63
6.8.1	Test Cases Against Use Cases	63
6.8.2	Other Test Cases	66
7	Conclusions	73
A	User Manual	75
A.1	How to search for a person using this software?	75
A.2	How to find all the indexed people in this software?	77

A.3	How can an admin login into the system?	78
A.4	How can an admin add a video source to the system?	80
A.5	How can an admin add a video to the system?	82
A.6	How can a super admin add a new admin?	84
A.7	How can someone be added as an indexed person to the system?	86
References		89