Criminal Face Detection

Mr.Shankar Nayak Bhukya
Associate Professor
Dept. Of Information Technology
Swami Vivekananda Institute Of Technology,Sec-Bad

Mr P.Karthik Reddy
Btech Final Year Student
Dept. Of Information Technology Line Swami
Vivekananda Institute Of Technology,Sec-Bad

Mr Kiran Kumar
Btech Final Year Student
Dept. Of Information Technology Line Swami
Vivekananda Institute Of Technology,Sec-Bad

Mr Kevin Ponnappa K.
Btech Final Year Student
Dept. Of Information Technology Line Swami
Vivekananda Institute Of Technology,Sec-Bad

1. Abstract:

Criminal record generally contains personal information about particular person along with photograph. To identify any Criminal we need some identification regarding person, which are given by eyewitness. In most cases the quality and resolution of the recorded image segments is poor and hard to identify a face. To overcome this sort of problem we are developing software. Identification can be done in many ways like finger print, eyes, DNA etc. One of the applications is face identification. The face is our primary focus of attention in social inters course playing a major role in conveying identify and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize face is remarkable.

2. INTRODUCTION:

2.1.PURPOSE OF THE PROJECT:

This project is aimed to identify the criminals in any investigation department. Here the technique is we already store some images of the criminals in our database along with his details and that images are segmented into many slices say eyes, hairs, lips, nose, etc. These images are again stored in another database record so to identify any criminals; eyewitnesses will see the images or slices that appear on the screen by using it we develop the face, which may or may not be matched with our images. If any image is matched up to 99% then we predict that he is only the criminal. Thus using this project it provides a very friendly environment for both operator and eyewitness to easily design any face can identify criminals very easy.

2.2. PROJECT OBJECTIVE:

This project is intended to identify a person using the images previously taken. The identification will be done according the previous images of different persons.

2.3. PROJECT SCOPE:

The scope of the project is confined to store the image and store in the database. When a person has to be identified the images stored in the database are compared with the existing details.

2.4. OVERVIEW OF THE PROJECT:

This project is aimed to identify the criminals in any investigation department. Here the technique is we already store some images of the criminals in our database along with his details and those images are segmented into many slices say eyes, hairs, lips, nose, etc. These images are again stored in another database record so to identify any criminals; eyewitnesses will see the images or slices that appear on the screen by using it we develop the face, which may or may not be matched with our images. If any image is matched up to 99% then we predict that he is only the criminal. Thus using this project it provides a very friendly environment for both operator and eyewitness to easily design any face can identify criminals very easy.

2.5. PROBLEM AREA DESCRIPTION

The project is aimed at identifying the criminals with the help of eye witness. There are mainly four modules in our project. They are Adding, Deleting, Updating and identifying the criminals. There are mainly three roles in our project. They are:

- Administrator

- Operator
- Eyewitness

The administrator is responsible for providing User id's and passwords. He provides authentication to the users. He creates deletes and Updates the User ids and Passwords.

The operator, who belongs to the investigating department, is responsible for entering the criminal details and maintains them. He adds, deletes and updates the criminal details. He also constructs the criminal face with the help of eye witness.

The eyewitness identifies the criminals with the help of cropped parts stored in a different database by the operator. The eyewitness selects a cropped part from the database and that cropped part will be freeze by the operator in this way, complete face of the criminal is constructed and the details of that criminal is retrieved from the database. We can also construct a new image from those cropped parts which we consider as an imaginary face of the criminal.

3. SYSTEM ANALYSIS

The first step in developing anything is to state the requirements. This applies just as much to leading edge research as to simple programs and to personal programs, as well as to large team efforts. Being vague about your objective only postpones decisions to a later stage where changes are much more costly.

The problem statement should state what is to be done and not how it is to be done. It should be a statement of needs, not a proposal for a solution. A user manual for the desired system is a good problem statement. The requestor should indicate which features are mandatory and which are optional, to avoid overly constraining design decisions. The requestor should avoid describing system internals, as this restricts implementation flexibility. Performance specifications and protocols for interaction with external systems are legitimate requirements. Software engineering standards, such as modular construction, design for testability, and provision for future extensions, are also proper.

Many problems statements, from individuals, companies, and government agencies, mixture requirements with design decisions. There may sometimes be a compelling reason to require a particular computer or language; there is rarely justification to specify the use of a particular algorithm. The analyst must separate the true requirements from design and implementation decisions disguised as requirements. The analyst should challenge such pseudo requirements, as they restrict flexibility. There may be politics or organizational reasons for the user requirements, but at least the analyst should recognize that these externally imposed design decisions are not essential features of the problem domain.

A problem statement may have more or less detail. A requirement for a conventional product, such as a payroll program or a billing system, may have considerable detail. A requirement for a research effort in a new area may lack many details, but presumably the research has some objective, which should be clearly stated.

Most problem statements are ambiguous, incomplete, or even inconsistent. Some requirements are just plain wrong. Some requirements, although precisely stated, have unpleasant consequences on the system behavior or impose unreasonable implementation costs. Some requirements seem reasonable at first but do not work out as well as the request or thought. The problem statement is just a starting point for understanding the problem, not an immutable document. The purpose of the subsequent analysis is to fully understand the problem and its implications. There is no reasons to expect that a problem statement prepared without a fully analysis will be correct.

The analyst must work with the requestor to refine the requirements so they represent the requestor's true intent. This involves challenging the requirements and probing for missing information. The psychological, organizational, and political considerations of doing this are beyond the scope of this book, except for the following piece of advice: If you do exactly what the customer asked for, but the result does not meet the customer's real needs, you will probably be blamed anyway.

3.1. Existing System:

This system is manual system only. Here, have a facility to store the criminal images. If you want to compare the criminal images with the existing images it is manual process. This process is very slow to give the result. It is very critical to find the criminal images.

3.2. Proposed System:

To overcome the drawbacks that were in the existing system we develop a system that will be very useful for any investigation department. Here the program keeps track of the record number of each slice during the construction of identifiable human face and calculate maximum number of slices of the similar record number. Based on this record number the program retrieves the personal record of the suspect (whose slice constituted the major parts of the constructed human face) on exercising the "locate" option.

3.3. Advantages:

- Very fast and accurate.
- No need of any extra manual effort.
- No fever of data loss.
- Just need a little knowledge to operate the system.
- Doesn't require any extra hardware device.
- At last very easy to find the criminals

3.4. Overview:

Addition,Clipping, onstruction and updating of the criminal record and face. Comparing the image with the faces that are there in our database. If any new images are found then it should be entered into our database by add image module and then it should be segmented into different slices.

4. SYSTEM DESIGN

During analysis, the focus is on what needs to be done, independent of how it is done. During design, decisions are made about how the problem will be solved, first at high level, then at increasingly detailed levels.

System design is the first design stage in which the basic approach to solving the problem is selected. During system design, the overall structure and style are decided. The system architecture is the overall organization of the system into components called subsystems. The architecture provides the context in which more detailed decisions are made in later design stages. By making high level decisions that apply to the entire system, the system designer partitions the problem into subsystems so that further work can be done by several designers working independently on different subsystems.

The system designer must make the following decisions:

- Organize the system into subsystems.
- Identify the concurrency inherent in the problem.
- Allocate subsystems to processors and tasks.
- Choose an approach for management of data stores.
- Handle access to global resources.
- Choose the implementation of control in software.
- Handle boundary conditions.
- Set trade-off priorities.

4.1. Breaking the System into Subsystems:

The first step in system design is to divide the system into small number of components. Each major component of a system is called a sub system. Each subsystem encompasses aspects of the system that share some common property – similar functionality, the same physical location, or execution on the same kind of hardware.

A subsystem not an object nor a function but a package of classes, associations, operations, events, and constraints that are interrelated and that have a reasonably well-defined and small interface with other subsystems. A subsystem usually identified by the services it provides. A service is a group of related functions that share some common purpose such as I/O processing. A subsystem defines a coherent way of looking at one aspect of the problem.

Each subsystem has a well-defined interface to the rest of the system. The interface specifies the form of all interactions and the information flow across subsystem boundaries but does not specify how the sub system is implemented internally. Each subsystem then can be designed independently without affecting the others.

The decomposition of systems into subsystems may be organized as a sequence of horizontal layers or vertical partitions.

A layered system is an ordered set of virtual worlds, each built in terms of the ones below it and providing the basis of implementation for the ones above it. The objects in each layer can be independent, although there is often some correspondence between objects in different layers. Knowledge is one-way only: a subsystem knows about the layers below it, but has no knowledge of the above layers. Each layer may have its own set of classes and operations. Each layer is implemented in terms of the classes and operations of lower layers.

Layered architecture comes in two forms: closed and open. In a closed architecture, each layer is built only in terms of the immediate lower layer. In a open architecture, a layer can use features of any lower layer to any depth.

We decomposed our system into three subsystems as layers. The three layers are closed architecture form. The three layers are GUI layer, Network layer and I/O layer. The purpose of GUI layer is to provide an efficient user interface to the user to interact with the system. It is built upon the Network layer which provides basic FTP services. The lowest layer is the I/O layer that provides services like reading or writing file to and from local and remote systems.

GUI subsystem
N/W subsystem
1) I/O subsystem

When top-level subsystems are identified, the designer should show the information flow among the sub systems. There are several architectural frameworks that are common in existing systems. They are batch transformation, continuous transformation, interactive interface, dynamic simulation, real-time system and transaction manager.

In the architectural frameworks specified above, our system will best suit in interactive interface architecture, since there are large number of interactions between system and user.

An interactive interface is a system that is dominated by interactions between the system and external agents, such as humans, devices or other programs. The external agents are independent of system, so their inputs can't be controlled, although the system may solicit responses from them.

4.2 Identifying Concurrency:

One important goal of system design is to identify which objects must be active concurrently and which objects have activity that is mutually exclusive. The latter objects can be folded together in a single thread of control or task. But there is no part that is concurrent in our system.

4.3 Allocating Subsystems to Processor:

In this step system designer estimates the hardware resources required and the implementation choice of either hardware or software. In our system all the subsystems will be implemented in software. The hardware requirements are general such as Pentium – III, 128 MB of RAM.

4.4 Management of Data Stores:

In this stage the system designer decides what format is used to store the data stores. There are DBMS systems or file systems and others. Here in our project there are no data stores except files. We then definitely prefer files to download and upload.

4.5 Choosing Software Control Implementation:

During the analysis, all interactions are shown as events between objects. But the system designer must choose among several ways to implement control in software. There are two kinds of control flows in a software system: internal and external. External control is the flow of externally visible events among the objects in the system. There are three kinds of control for external events: procedure driven, event driven sequential and concurrent. Internal control is the flow of control within a process.

Our system falls in external control flow like event driven control. Events are fired by external agent such as user in different order but sequential not concurrent.

5. Testing

5.1 Testing Introduction:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software testing can also be stated as the process of validating and verifying that a software program/application/product:

1. meets the business and technical requirements that guided its design and development;
2. works as expected; and
3. Can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

5.2 Test Cases

1. First of all check whether it is taking the real Human Face or it also authenticates for the dummies.
2. Check the distance up to which it accepts the input of the features of the human correctly.
3. It must be noted that it should take into account the smallest features of the face like nose, ear, eyes etc. because beard can grow and it becomes difficult to authenticate for the person
4. If the laser has been used the rays should not be harmful it should be in the limit.
5. It should be able to authenticate the person in any of the sitting postures of that person no matter where the person is sitting or looking at.

Conclusion:

Criminal Face Detection System is a challenging problem in the field of image analysis and computer vision that has received a great deal of attention over the last few years because of its many applications in various domains. In this project, we have presented the implementation of and robust and efficient face recognition system in JAVA environment. We have demonstrated the general steps that could be used for its implementations. Our implemented system is customized for its applications into criminal face detection system however the generalized steps can be customized to adapt to various applications in various different environments. We have also listed the techniques which can be deployed in JAVA environment in this domain.

References

[1] M. Abdel-Mottaleb and A. Elgammal, Face detection in complex environments, in Proceedings International Conference on Image Processing, 1999.

[2] Albiol, A simple and efficient face detection algorithm for video database applications, in Proceedings of the 2000 International Conference on Image Processing, 2000, p. TA07.09.

[3] Albiol, C. A. Bouman, and E. J. Delp, Face detection for pseudosemantic labeling in video databases, in Proceedings International Conference on Image Processing, 1999.

[4] Y. Amit, D. Geman, and B. Jedynak, Efficient focusing and face detection, in Face Recognition: From Theory to Application, Springer-Verlag, Berlin/New York, 1998.
[5] P. J. L. Van Beek, M. J. T. Reinders, B. Sankur, and J. C. A. Van Der Lubbe, Semantic segmentation of videophone image sequences, in Proc. of SPIE Int. Conf. on Visual Communications and Image Processing, 1992, pp. 1182–1193.
[6] S. Ben-Yacoub, B. Fasel, and J. L¨uttin, Fast face detection using MLP and FFT, in Proceedings Second International Conference on Audio- and Videobased Biometric Person Authentication (AVBPA), 1999.
[7] D. E. Benn, M. S. Nixon, and J. N. Carter, Extending concentricity analysis by deformable templates for improved eye extraction, in Proceedings Second International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA), 1999.