BFCAI
Faculty of Computers and Artificial Intelligence
Computer Science Department

Human-Computer Interaction

(HCI)

A. I. M. Shaban, T. G. M. Hassan, S. A. A. Mohamed, and O. A. S. Abdelhamid, "Human-Computer Interaction (HCI)," supervised by Dr.Sh.sarhan , Faculty of Computers & Artificial Intelligence (BFCAI), Egypt.

Abstract

Human-computer interaction (HCI) principles are foundational in the development of intelligent systems, guiding design from conception to execution. These principles, ranging from user-centric design to insights borrowed from cognitive engineering, shape user behavior and minimize blind spots in the design process. By adhering to HCI principles, products and experiences are crafted to resonate with users across various domains, including AI-powered applications such as autonomous driving, AI-generated media, personalized medicine, and virtual reality gaming. The establishment of user trust in these systems is paramount, as it influences their acceptance and utilization. This paper explores the significance of HCI principles in the context of intelligent system design, focusing on emerging interaction paradigms, user interface design, and foundational principles. Through this exploration, the aim is to enhance user-technology interaction, making it more intuitive, inclusive, and efficient.

Research question

How can the integration of user-centered design principles , through user research, and iterative prototyping methods improve the usability and satisfaction of interactive systems?

How can we leverage user-centered design principles, user research, and iterative prototyping methods to enhance the user experience in interactive systems?

Introduction

Human-computer interaction (HCI) has evolved significantly since its inception in the early 1980s, initially emerging as a specialized area within computer science that integrated insights from cognitive science and human factors engineering. Over the past three decades, HCI has experienced rapid expansion, drawing professionals from diverse disciplines and incorporating a wide range of concepts and methodologies. Today, HCI represents a convergence of various fields within human-centered informatics, reflecting a synthesis of disparate approaches to research and practice.

The origins of HCI can be traced back to the late 1970s when computers were primarily accessed by IT professionals and enthusiasts. However, the landscape changed dramatically with the advent of personal computing, which democratized access to computers and underscored their usability shortcomings for everyday users. The proliferation of personal software and computer platforms transformed individuals worldwide into potential computer users, accentuating the need for improved usability in computer interfaces and applications.

Background & Related work (Literature Review)

The literature on Human-Computer Interaction (HCI) provides valuable insights into its evolution and significance since its inception in the early 1980s. Here are key themes from the literature:

Historical Perspectives: Scholars have explored the historical roots of HCI, tracing its origins to the pioneering work of researchers such as Douglas Engelbart and Alan Kay in the 1960s and 1970s. These early endeavors laid the foundation for the development of user-centered design principles and interactive computing systems.

Interdisciplinary Collaboration: HCI literature emphasizes the importance of interdisciplinary collaboration, highlighting the contributions of cognitive science, psychology, sociology, and ergonomics to HCI theory and practice. Studies have examined how insights from these diverse disciplines have shaped the design and evaluation of interactive systems.

Methodological Advancements: The HCI literature documents methodological advancements in research methods and techniques for studying user behavior and designing user interfaces. This includes usability testing methodologies, task analysis techniques, and participatory design approaches that prioritize user involvement throughout the design process.

Technological Innovations: HCI research has explored technological innovations that have revolutionized user interaction with computers, such as graphical user interfaces, touchscreens, and mobile devices. Studies have examined the impact of these technologies on user experience and interaction design principles.

User Experience (UX) Design: A significant focus of HCI literature is on UX design, which emphasizes the holistic experience of users interacting with technology. Research in this area explores methods for enhancing usability, accessibility, and emotional engagement in interactive systems.

Inclusive Design Practices: HCI literature addresses the importance of inclusive design practices that consider the needs and preferences of diverse user populations, including individuals with disabilities, older adults, and people from different cultural backgrounds.

Ethical Considerations: HCI research examines ethical considerations related to user privacy, data security, and algorithmic bias in interactive systems. Studies have investigated ethical frameworks and guidelines for designing and evaluating technology in a responsible and socially conscious manner.

2.1.1 Technology:

Using Django web framework that follows the model-view-controller (MVC) architectural pattern. It provides a robust set of tools and features for building web applications quickly and efficiently. Django follows the principle of "Don't Repeat Yourself" (DRY), promoting code reusability and maintainability. It is known for its scalability, security, and ease of use, making it a popular choice for web development.

Django provides a structured environment for deploying machine learning models as part of a web application. You can encapsulate your trained models within Django views or custom API endpoints, allowing users to interact with your models via a web interface or API requests.

HAAR cascade:

HAAR cascade is a simple and efficient model based on Haar features. It detects faces by first identifying large features like eyes and noses, then progressively detecting smaller features like the mouth and chin.

Benefits: HAAR cascade is fast and computationally efficient, making it suitable for real-time face detection applications. It can achieve decent accuracy in detecting faces.

Faster R-CNN:

Faster R-CNN is a more complex face detection model that uses a region proposal network (RPN) to generate candidate regions for face detection. It then classifies each candidate as a face or non-face.

Benefits: Faster R-CNN is more accurate than the HAAR cascade, but it is also more computationally expensive. It performs well in scenarios where accuracy is a priority and computational resources are available.

SSD (Single Shot MultiBox Detector):

SSD is a face detection model that utilizes anchor boxes of different scales and aspect ratios to predict offsets and confidence scores for each anchor box. It produces predictions at multiple scales and explicitly separates predictions by aspect ratio.

Benefits: SSD is accurate like R-CNN but faster. It saves time by detecting objects in a single pass. SSD is suitable for applications where both accuracy and speed are important.

Xception:

Xception is a deep convolutional neural network architecture that utilizes depthwise separable convolutions. It is often used for face recognition tasks.

Benefits: Xception achieves state-of-the-art results in various computer vision tasks, including image classification, object detection, and segmentation. It is efficient due to the use of depthwise separable convolutions.

FaceNet:

FaceNet is a deep CNN architecture commonly used for face recognition. It has a large number of parameters and requires significant computational resources.

Benefits: FaceNet is known for its simplicity and accuracy in face recognition tasks. It can achieve high performance with a large amount of labeled training data.

VGG (Visual Geometry Group):

VGG is a deep CNN architecture known for its large number of layers and parameters.

Benefits: VGG models excel in capturing detailed features and have achieved excellent performance in various computer vision tasks. However, their larger size can pose challenges in terms of deployment and computational resources.

DeepFace:

DeepFace is a deep neural network architecture designed for face recognition.

It has been trained on a large-scale dataset.

Benefits: DeepFace can achieve high accuracy in face recognition tasks. However, it may be sensitive to variations in pose, lighting conditions, and occlusions. Interpretability of its complex architecture may also be a challenge.

Eigenfaces:

Eigenfaces is a simple and effective face recognition algorithm commonly used in applications.

Benefits: Eigenfaces algorithm is straightforward and can provide decent face recognition results. However, it may be sensitive to variations in poses and expressions.

LBPH (Local Binary Patterns Histograms):

LBPH is a face recognition algorithm that is more robust to pose and expression variations compared to Eigenfaces.

Benefits: LBPH algorithm can handle variations in poses and expressions to a certain extent. While it may not be as accurate as deep learning algorithms, it can be a viable option in some scenarios.

The Proposed Solution

Model Methodology:

.1 Data Collection:

We trained our model on a subset of the Extracted faces data set, which we first describe. We then provide details with respect to verification and one-shot performance. The dataset contains 1324 different individuals, with 2-50 images per person.

The images are of size (128,128,3) and are encoded in RGB (Red, Green, Blue).

Each folder and image are named with a number, i.e., 0.jpg, 1.jpg

Length of training list: 1191 Length of the testing list: 133

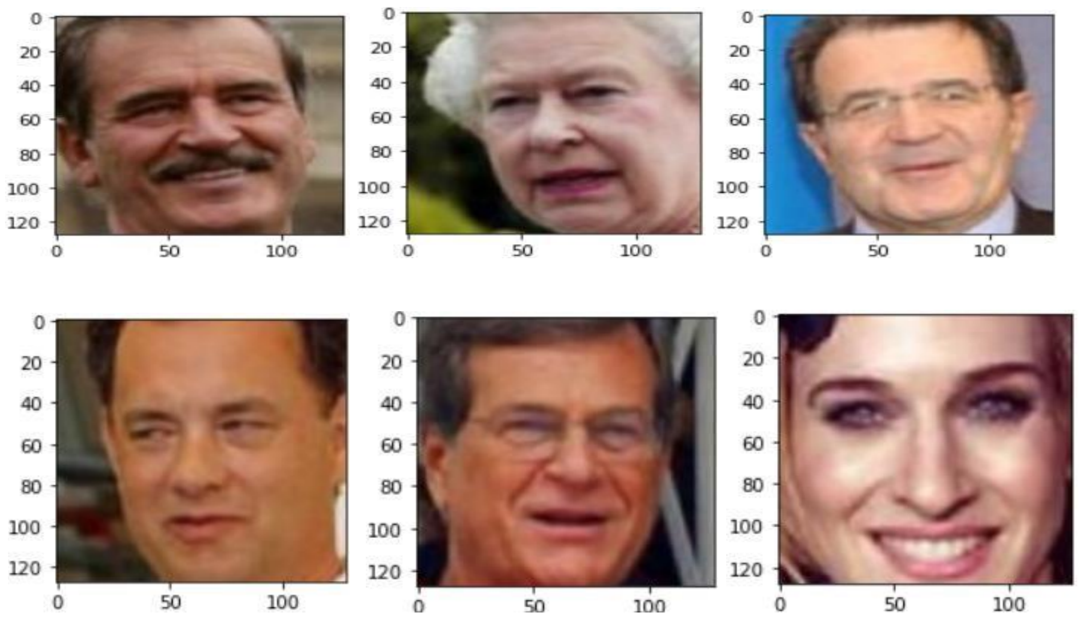

Figure1.1:the arrangement and labeling suggest that the dataset comprises image of face used for training and teasting purposes in a machine learning model.

2 Data Processing:

The preprocess_input function takes an image as input and returns a NumPy array that is in the format expected by the Xception model.

We use the train and test list to create triplets of (anchor, positive, negative) face data, where positive is the same person and negative is a different person than an anchor.

Number of training triplets: 9996 Number of testing triplets: 1235

Here is the sample for the triplet:

Figure2.1: facial recognition algorithm, illustrating the concept of triplet loss where each triplet contains an anchor image, a positive image (same person as the anchor), and a negative image (different person from the anchor).

.3 Model selection:

These are the basic operations involved in Face Recognition system:

Face Detection: It's the first and most essential step in face recognition.

We use the single shot detector (SSD) as a detection model for detecting the face from the captured image Either single or multi-face.

The second step is the Face Recognition model. It is less reliable, and the accuracy rate is still not up to the mark. Extensive work on Face Recognition has been done, but still, it is not up to the mark from an implementation point of view

We used for this task Convolution Siamese neural network (CNN + Siamese Network) that gets the similarity between the input image inside the database and the input image.

We use a pretrained model (Xception) using the last 27 layers and connected using 2 fully connected layers.

4.4 Model Training:

We train our model on the dataset Forword and backward with 30 epochs.

We choose the triplet loss which compares three samples: an anchor sample, a positive sample, and a negative sample.

The weights to use for the model is ‘ImageNet’ which will load the pre-trained weights for all networks that were trained on the ImageNet dataset.

.5 Model evaluation:

We get the accuracy of the model :92%

We test 994 sample and

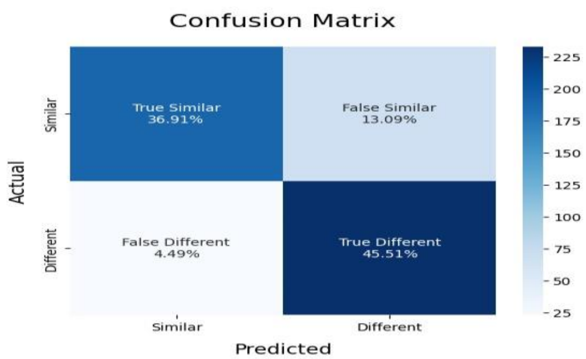

Figure5.1: the confusion matrix provides a way to visualize the performance of a classification algorithm.

6.Proposed model:

1 For face detection:

HAAR cascade:

The HAAR cascade is a simple and efficient model that is based on Haar features. Haar features are small rectangular features that are used to detect faces in images. The HAAR cascade works by first detecting large features, such as eyes and noses. If these features are detected, the cascade then proceeds to detect smaller features, such as the mouth and chin. The HAAR cascade is a fast and the less accurate model.

Faster R-CNN:

Faster R-CNN is a more complex model than the HAAR cascade. Faster R-CNN uses a region proposal network (RPN) to generate candidate regions for face detection. The RPN is then used to classify each candidate region as a face or non-face. R-CNN is more accurate than the HAAR cascade or CNN, but it is also more computationally expensive, R-CNN does not have the high speed as SSD.

SSD:

The model uses a set of anchor boxes, which are predefined bounding boxes at different scales and aspect ratios that cover the range of face locations and sizes.

The model predicts offsets and confidence scores for each anchor box, which are used to generate final bounding boxes around detected faces,he is accurate as same as RNN, it saves a lot of time.

SSD model produces predictions at different scales from the feature maps of different scales and explicitly separates predictions by aspect ratio.

2. For face recognition:

Xception:

It is a modification of the Inception CNN architecture that uses depthwise separable convolutions instead of traditional convolutions.

Depth wise separable convolutions are a type of convolution that performs two operations in one: a depth wise convolution and a pointwise convolution. The depth wise convolution operates on each channel of the input feature map independently, while the pointwise convolution operates on all channels of the input feature map together.

This makes depthwise separable convolutions more efficient than traditional convolutions, as they require fewer parameters and computations. Additionally, depthwise separable convolutions are more effective at learning spatial features than traditional convolutions.

Xception has been shown to achieve state-of-the-art results on a variety of computer vision tasks, including image classification, object detection, and segmentation.

FaceNet:

Deep CNN architecture is known for its simplicity and accuracy.

Computationally expensive: FaceNet utilizes a deep neural network architecture with a large number of parameters, making it computationally intensive and requiring significant computational resources.

Training data requirements: FaceNet typically requires a large amount of labeled training data to achieve optimal performance, which can be challenging to obtain in certain scenarios.

Sensitivity to variations: FaceNet may be sensitive to variations in pose, lighting conditions, and occlusions, which can affect its performance in real-world scenarios.

VGG:

Large model size: VGG models have a relatively enormous number of layers and parameters, resulting in a larger model size. This can make it challenging to deploy the model on resource-constrained devices or in applications with limited storage capacity.

Limited receptive field: VGG models utilize small kernel sizes (3x3) in each convolutional layer, which limits their ability to capture larger spatial context and global features.

DeepFace:

Training data bias: DeepFace has been trained on a large-scale dataset that may introduce biases present in the data, leading to potential limitations in recognizing faces from diverse demographics or underrepresented groups.

Lack of interpretability: DeepFace's deep neural network architecture is complex, making it difficult to interpret and understand the learned representations and decision-making processes.

Deep learning in detail:

As you make an attendance system, we use a face recognition deep learning model, that takes two model face detection and face recognition model.

Attendance takes place by taking the image of one or more people to take the attendance by verify in the image in the images in database.

Face detection is to detect the faces in the image, face recognition is to ask who is in that image.

- Face detection model detects and localizes the face in the input image.

The SSD model architecture used is based on a deep convolutional neural network (CNN) with multiple layers that process the input image to extract features at various levels of abstraction.

The model uses a set of anchor boxes, which are predefined bounding boxes at different scales and aspect ratios that cover the range of possible face locations and sizes.

The model predicts offsets and confidence scores for each anchor box, which are used to generate final bounding boxes around detected faces.

iterates over the detected faces and selects the first one with a confidence score above a certain threshold.

The selected face is then cropped from the input image and resized to a fixed output size of (128, 128).

The SSD model detects objects in a single pass, which means it saves a lot of time. But at the same time, the SSD model also has amazing accuracy in its detection.

The VGG-16 network is used as the backbone for feature extraction. Only the layers from Conv1_1 to Pool5 are retained, and all other layers are removed. The backbone network captures hierarchical features from the input image.

The fully connected layers fc6 and fc7 in VGG-16 are converted to convolution layers (Conv_fc6 and Conv_fc7) by subsampling the parameters. After Conv_fc6 and Conv_fc7, four additional convolution layers are added: Conv6_1, Conv6_2, Conv7_1, and Conv7_2. These convolution layers extract more complex features from the backbone network.

L2 normalization is applied to the feature maps of Conv3_3, Conv4_3, and Conv5_3 to scale the norms of the features to 20. This normalization helps in handling feature maps with different scales and improves detection accuracy.

Deconvolution layers are used to expand the high-level feature maps to match the dimensions of the low-level feature maps. This expansion allows the integration of large-scale context information, which helps improve the detection accuracy. The high-level feature maps are summed with the corresponding low-level feature maps in an element-wise manner.

Six feature layers are used as detection layers after the deconvolution operation (D1, D2, D3, D4, D5, and D6). These detection layers predict the locations and sizes of faces. Each detection layer is followed by a convolution layer with a kernel size of $3\times33\times3$ to generate the final detection results. The detection layers have different scales and are associated with different scales of anchor boxes.

For each anchor box, the detection layers predict four offsets related to its coordinates (for localization) and C+1 scores for classification. The number of labels, C, depends on the detection layer, with $C\geq1$ for the D1 detection layer (using a max-out background strategy) and $C=1$ for other detection layers.

It uses the feature pyramid network to make better detection.

The feature pyramid network store first appeared with high resolution but having poor semantic value unlike the in Forword that reduce the resolution with rich semantic value.

We take the last layer in each level and compute elementwise operation with the deconvolution layer to get the detector layer to make multiscale for each level you make this operation.

- Face recognition solve the problem Laking of data by using technique called oneshot learning.

We have two Approach for classifying images first one using multi-class classification and the other One-shot learning.

One-shot learning is a machine learning technique that aims to recognize or classify objects based on a single example or a small number of examples. Unlike traditional machine learning approaches that require enormous amounts of labeled data for training, one-shot learning focuses on learning from limited information.

We use the Siamese convolution neural network that helps in technique one-shot learning as this is used in 2 embedding architecture and connecting with similarity layer.

The input image to the architecture must be size 128*128*3 where the number 3 refer to the image must be 3 color channels (RGB) not gray scale.

The embedding initialized weight by ‘image net’.

The embedding architecture contains filters 3*3 and 1*1 convolution layer to extract features from the input image.

1x1 convolutions are primarily used for dimensionality reduction. reduce number of channels in feature map without loss any information, that help to learn complex features.

3x3 convolutions are primarily used for feature extraction. This is because they can be used to learn more complex features from the input data. This can be useful for tasks such as image classification and object detection.

The embedding uses depthwise separable convolutions instead of regular convolutions. Depthwise separable convolutions are a type of convolution that divides the convolution operation into two steps: a depthwise convolution and a pointwise convolution. The depthwise convolution operates on each input channel independently, while the pointwise convolution operates on all of the output channels together. This makes depthwise separable convolutions faster and more efficient than regular convolutions, while still maintaining accuracy.

Each of which contains a depthwise separable convolution followed by an average pooling layer, which will apply global average pooling to the output of the model.

The modules are stacked together in a residual connection fashion, which means that the output of each module is added to the input of the next module. This helps improve the network's accuracy by preventing information loss during convolution operations.

The architecture is connected with 2 fully connected layers 512 then 256 which produces a 1-D tensor of shape.

Each fully connected layer has a normalization layer to normalize the output from them.

The basic idea behind triplet loss is to compare three samples: an anchor sample, a positive sample, and a negative sample. The anchor sample is the reference point, the positive sample is a sample that should be more similar to the anchor, and the negative sample is a sample that should be dissimilar to the anchor.

$L = \max(d(a, p) - d(a, n) + margin, 0)$ where:
$d(a, p)$ represents the distance between the anchor (a) and the positive sample (p) in the embedding space.

$d(a, n)$ represents the distance between the anchor (a) and the negative sample (n) in the embedding space.

margin is a hyperparameter that specifies a minimum desired separation between the positive and negative distances. It ensures a margin of difference between the positive and negative pairs.

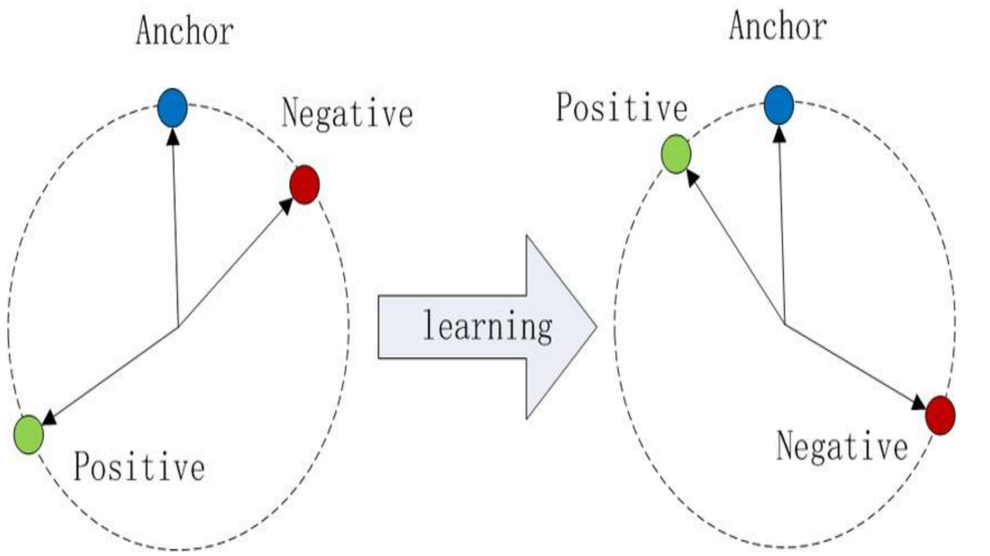

Figure6.2.1:diagram illustrating positive and negative anchors in a machine learning context.

7.The model overviews

Object detection model

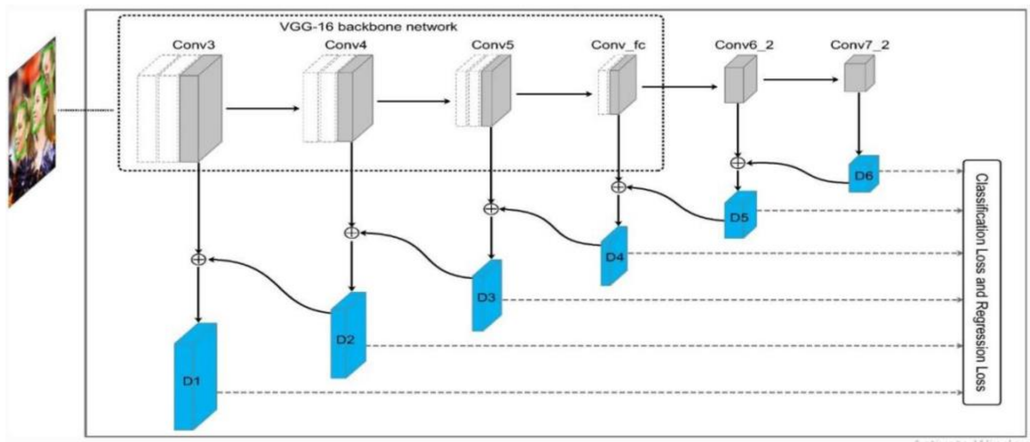

Figure7.1:a computer screen with a diagram of a computer network.

we retain the layers from Conv1_1 to Pool5 in the VGG-16 network and remove all other layers. Then, we add several auxiliary structures, i.e., the additional convolution layers, the normalization layers, the deconvolution layers, and the detection layers, to adapt the backbone network for the purpose of face detection.

The additional convolution layers are used to increase the feature map size and improve the detection accuracy for small faces.

The normalization layers are used to normalize the features and improve the stability of the training process.

The deconvolution layers are used to unsampled the features and integrate largescale context information.

The detection layers(D1,D2,.) are used to predict the locations and sizes of faces.

The detection layers are arranged at multiple scales and aspect ratios to cover a wide range of possible face sizes and shapes. For each anchor, the SSD predicts four offsets related to its coordinates and C+1 scores for classification, where C denotes the number of classes.

The offsets are used to adjust the coordinates of the anchor to the actual location of the face. The scores are used to determine the probability that the anchor contains a face of a particular class.

7.2 face recognition model:

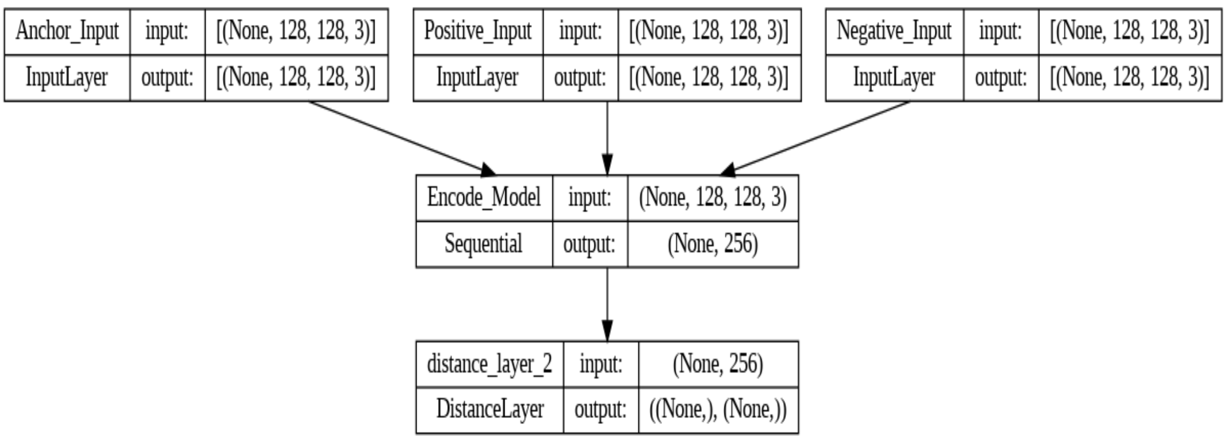

Figure7.2 :The flowchart appears to visualize a facial recognition model. Text labels include "Anchor Input," "Positive_Input," "Negative_Input" and "Encode Model."

The encode model:

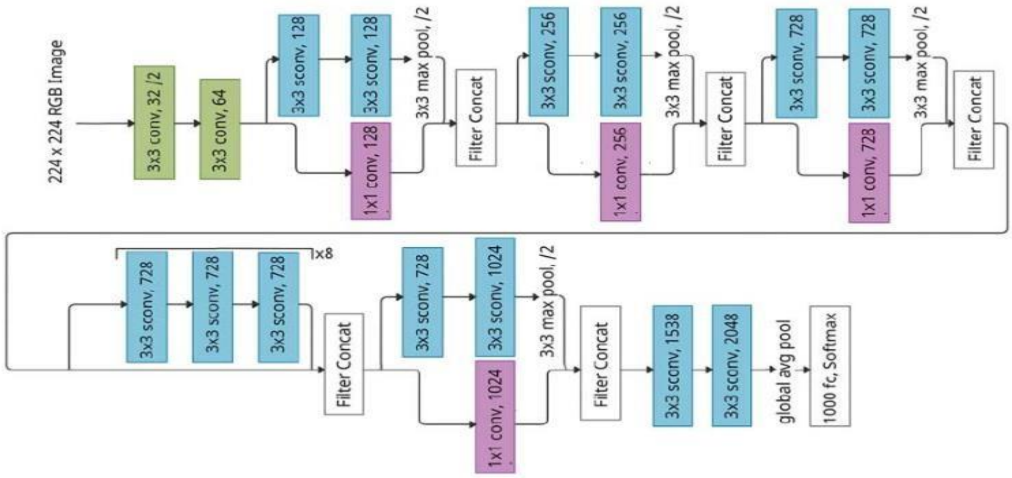

Figure 2: Architecture of the Xception deep CNN model

In the Xception model we freeze the first 41 layer and use the last 27 layer

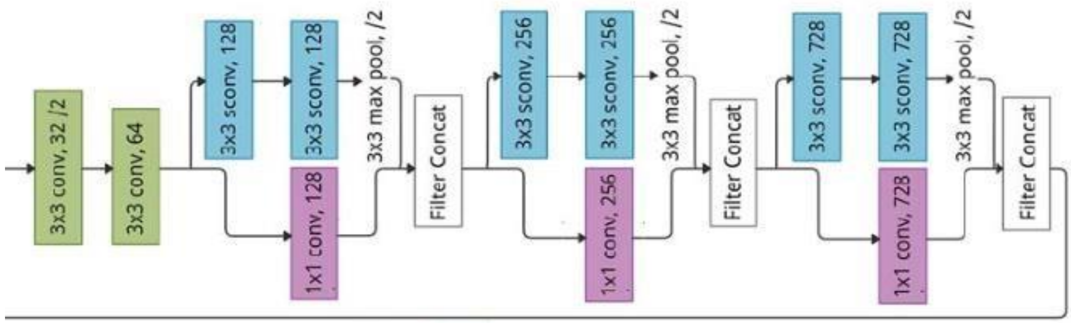

Figure7.4:The rightmost side of the image shows a layer labeled “Filter Concat,”

Entry Flow: This is the initial part of the network responsible for extracting lowlevel features from the input image. It consists of a series of convolutional layers, including depthwise separable convolutions. The goal is to capture local information efficiently.

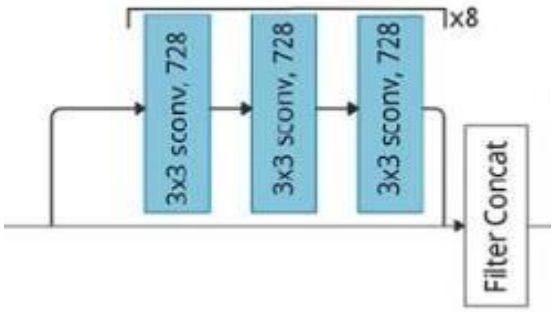

Figure7.5: “8XL” is located in the top left corner of the diagram. The remaining part of the diagram consists of three rows, each containing a blue filter with a grid of weights.

Middle Flow: This part of the network is composed of multiple residual blocks stacked together. Each residual block consists of several depthwise separable convolutions, which help in preserving and propagating useful information through skip connections. The skip connections aid in gradient flow and enable the model to learn more effectively.

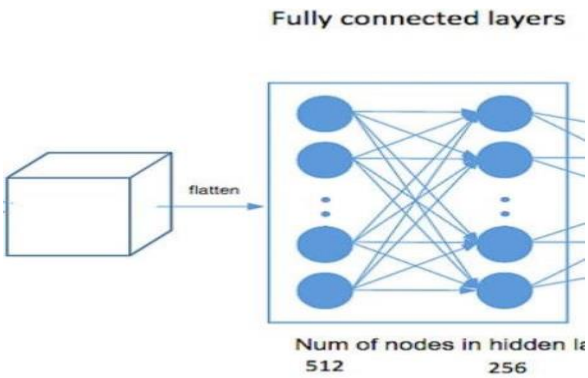

Figure7.6: depicts a visualization of a convolutional neural network (CNN) filter. CNNs are a type of artificial neural network commonly used for image recognition and classification tasks.

After freeze the last layer of pretrained model we use flatten layer

Flatten: flattens the output of the previous layer, converting it from a 2D tensor into a 1D tensor. The output shape is (None, 2048), which means it has 2048 elements.

This is a fully connected (dense) layer with 512 units. It takes the flattened input and produces a tensor of shape (None, 512). The number of parameters in this layer is 1,049,088.

Add Batch Normalization layer performs batch normalization on the inputs. It helps in normalizing the activations of the previous layer, making the training process more stable. The output shape remains the same as the input, (None, 512).

Another fully connected layer with 256 units. It takes the output of the previous layer and produces a tensor of shape (None, 256). The number of parameters in this layer is 131,328.

Lambda layer represents a lambda layer, which allows you to define and apply custom operations to the input data. The specific operation performed by this lambda layer is not provided in the model summary.

7.Comparator of models.

7.1 For face detection:

Aspect	HAAR Cascade	Faster R-CNN	SSD
Detection Approach	Uses Haar features to detect faces in images.	Uses a Region Proposal Network (RPN) to generate candidate regions.	Uses anchor boxes at different scales and aspect ratios.
Key Features	- Simple and efficient. - Detects large features first (eyes, noses) and then smaller features (mouth, chin).	- Complex and accurate. - Uses RPN to classify regions as face or non-face.	- Accurate and fast. - Predicts offsets and confidence scores for anchor boxes.
Accuracy	Less accurate	More accurate than HAAR cascade and CNN	Comparable to Faster R-CNN in accuracy
Speed	Fast	Slower due to computational complexity	High speed, saves a lot of time
Computational Expense	Low	High	Moderate
Applications	Quick face detection in real-time applications	High-accuracy face detection where computational resources are available	Real-time face detection in resource-constrained environments
Model Complexity	Low	High	Moderate

Table 1: Describe the difference between the models in face detection .

7.2. For face recognition:

Aspect	Xception	FaceNet	VGG	DeepFace
Architecture	Depthwise separable convolutions, modification of Inception	Deep CNN	Deep CNN with many layers	Deep CNN
Efficiency	High efficiency due to depthwise separable convolutions	Computationally expensive	Computationally	

Table 2: Describe the difference between the models in face recognation

Implementation, Experimental Setup, & Results

The training and loss curve:

The Trian accuracy of the model face recognition:

99% the

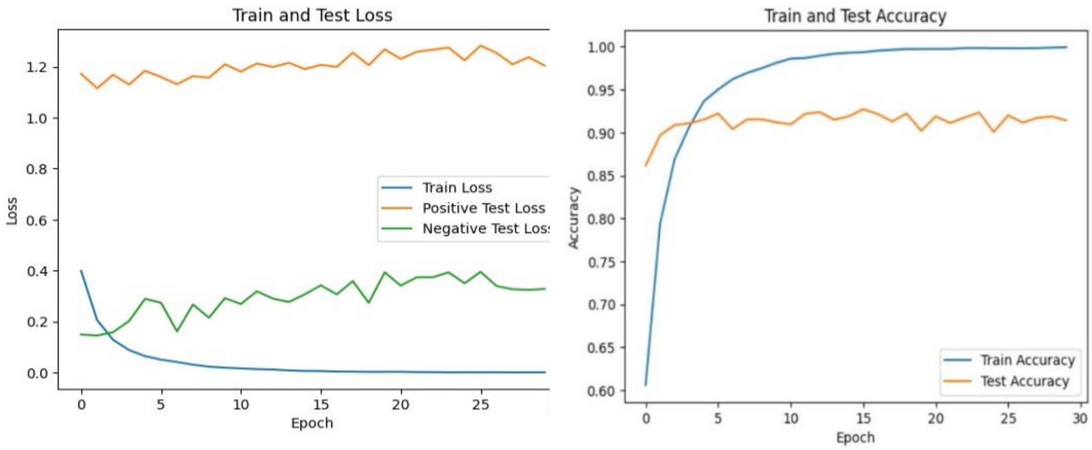

Figure7:7The loss of the test positive and the test negative are suitable for its solution.

And test accuracy of the model face recognition: 92%

The discussion:

various challenges and issues in face recognition algorithms and highlights different areas of research and development to address these challenges. Here is a breakdown of the main points:

Cross Factor Face Recognition: This involves dealing with variations in pose, age, and makeup in face recognition. Techniques such as frontalization, generative probabilistic models, and matching makeup and non-makeup faces are used to address these challenges.

Heterogeneous Face Recognition: This deals with recognizing faces across different modalities such as near infrared and visual spectrum images, lowresolution images, and photo sketches. Approaches like matching visible and near-infrared images, deep learning for low-resolution faces, and image-to-image translation for photo-sketch matching are mentioned.

Single or Multiple Media Face Recognition: This category includes low-shot face recognition, template-based face recognition, and video face recognition. Approaches like training data enlargement, feature learning, similarity comparison, max pooling, and video face representation are used to improve performance in these scenarios.

Industry-Based Face Recognition: This focuses on specific applications and challenges in face recognition, such as 3D face recognition, partial face recognition, face anti-spoofing, and mobile device-based face recognition. Methods like 3D data synthesis, deep feature extraction, arbitrary-size face patches, anti-spoofing techniques, and lightweight recognition for mobile devices are mentioned.

Covariates and CNNs (Convolutional Neural Network): Convolutional Neural Networks (CNNs) are widely used in deep face recognition systems. The passage mentions the impact of various covariates (e.g., image quality factors, model characteristics) on CNN performance. Factors like noise, blur, missing pixels, and brightness are found to significantly affect performance, while changes in contrast and compression artifacts have a lesser impact. Computation strategies for descriptors and colour information are not found to have a significant effect on performance.

Conclusion and Future

Overview:

The education system is one of the most important systems in now days and detecting its problem and improving its capability and functionality is an important goal to make the system more accurate and more usable. the technology now is playing an important role in solve our problem without the need of interact manually only by technology we can produce smarter solutions and, in our problem, here it's the best decision to achieve the proposed goal

Conclusion:

The main problem that this project done for is to end the suffering of using manually attendance system that may be confusing because the number of students is more growing and the time that the teacher takes to record attendance to the system and the unethical attendance by some of the student. we used deep learning to beat this issue by making digital attendance system by face recognition and detection Used by teacher and student in different ways

The teacher uses the attendance system by attaching a photo of the student in an offline lecture and the system will detect faces and recognizes to each student by comparing his photo to the database and make a report of the attendance to rate each student

The student uses the attendance system if he is in an online lecture by livestream camera to recognize his face directly and send his attendance to the database and then to the teacher to make the report this way is to ensure that attendance taken in ethical way.

The digital attendance system based on deep learning methodologies using simese model to face recognition and cv2 caffe model to multi face detection. they take a preprocessed photos and train on it to learn how to make difference between photos and decide if this is the student or not.

This is the main idea of the project and the education web page is for covering it and make reason to use it .

Comparison of the References :

Criteria	Hartson (1998)	Bansal & Khan (2018)	Miraz, Ali, & Excell (2021)	Milani et al. (2024)
Title	HCI: Interdisciplinary roots and trends	Review Paper on HCI	Adaptive UIs and usability through UI plasticity	Systematic Review of HCI Research
Objectives	Review HCI roots and trends	Review recent HCI developments and challenges	Explore adaptive UIs and usability	Review HCI in medical and engineering fields
Methodology	Historical and theoretical review	Literature review	Theoretical analysis	Systematic review
Accuracy	Dependent on source quality; historical	Relies on review thoroughness and source quality	Based on theoretical framework robustness	High due to systematic and comprehensive approach
Strengths	Broad historical context, interdisciplinary	Focus on recent developments and challenges	Emphasizes adaptability and inclusivity	Detailed insights into HCI applications
Weaknesses	Lack of empirical validation, may be outdated	May lack depth and comprehensive context	Primarily theoretical, limited empirical data	Limited to specific applications, may miss broader trends
References	Wide range from multiple disciplines	Recent literature on modern trends	Contemporary research on adaptive UIs	Extensive studies from medical and engineering journals

Table 3: Comparison of Selected Human-Computer Interaction (HCI) Research Papers

Futures

1-A portable fingerprint device has been developed that may be circulated among students to allow them to lay their finger on the sensor during lecture time without the intervention of the instructor. This technology ensures that attendance is recorded in an error-free manner.

2- voice recognition for student who cannot share their photos to the system then the voice is the solution of this problem to take an attendance

REFERENCES

H. R. Hartson, "Human–computer interaction: Interdisciplinary roots and trends," *J. Syst. Softw.*, vol. 43, no. 2, pp. 103-118, Nov. 1998.

H. Bansal and R. Khan, "A Review Paper on Human Computer Interaction ," *Int. J. Comput. Sci. Eng.*, vol. 8, no. 4, Apr. 2018.

M. H. Miraz, M. Ali, and P. S. Excell, "Adaptive user interfaces and universal usability through plasticity of user interface design," *J. Comput. Sci.*, vol. 40, 100363, May 2021 .

A. S. Milani, A. Cecil-Xavier, A. Gupta, J. Cecil, and S. Kennison, "A Systematic Review of Human–Computer Interaction (HCI) Research in Medical and Other Engineering Fields," *Int. J. Hum.-Comput. Interact.*, vol. 40, no. 3, pp. 515-536, 2024. doi: 10.1080/10447318.2022.2116530 .

Preece, J., Rogers, Y., & Sharp, H. (2015). *Interaction design: Beyond human-computer interaction.* John Wiley & Sons.

A comprehensive guide to interaction design, covering principles, methods, and case studies in HCI.

Carroll, J. M. (Ed.). (2012). *Human-computer interaction in the new millennium.* Addison-Wesley.

This edited volume offers insights from leading HCI researchers on emerging trends and future directions in the field.

Rogers, Y., Sharp, H., & Preece, J. (2011). *Interaction design: Beyond human-computer interaction.* John Wiley & Sons.

An updated edition of the classic textbook on interaction design, covering contemporary topics such as mobile and ubiquitous computing.

Bødker, S., & Anderson, K. M. (Eds.). (2020). *Human-computer interaction: Interact '90.* Elsevier.

This collection of papers from the Interact '90 conference provides historical insights into HCI research and development.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction.* CRC Press.

A foundational text that explores the psychological principles underlying human-computer interaction.

Dix, A., & Others. (2009). *Human-computer interaction.* Springer.

This textbook offers a comprehensive overview of HCI concepts, theories, and methodologies, suitable for students and researchers alike.

Jyoti, & Kaur, G. (2023). Research Paper on Human Computer Interaction (HCI). International Journal for Multidisciplinary Research (IJFMR), 5(2). Retrieved from https://www.ijfmr.com/