International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Gender Classification Using Convolutional Neural Network (CNN)

Cherukuri Vyshnavi, Nookala Sai Homitha, Balabhadra Vasavi, Mareedu Bhavana, Suneetha Bulla

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: The purpose of this paper is to demonstrate an innovative convolutional neural network (also known as CNN) methodology for real-time categorization of gender via face photos. The suggested CNN architecture boasts much reduced computational complexity than the current methodologies used in pattern recognition applications. By combining convolutional and subsampling layers, the overall processing layer count is minimised to four. Notably, using cross-correlation versus standard convolution tends to alleviate computing strain. The association is programmed using extended worldwide acquisition frequencies and a second-order backpropagation learning algorithmic framework. The demonstrated CNN approach has been examined using two freely downloadable facial statistics, SUMS and AT&T, with classification accuracies of 99.38% and 98.75%, respectively. Furthermore, the neural network's algorithm demonstrates exceptional efficiency by analysing and categorising a 32 by 32-pixel face picture in just 0.27 milliseconds, resulting in an outstanding consumption of over 3700 images per second. The successful performance of the proposed CNN methodology is further demonstrated by its speedy convergence throughout the training process, which requires less than 20 epochs. The results were produced to showcase the suggested CNN's outstanding accuracy in accurately categorising data, launching it as a realistic and effective real-time identification of the gender system.

Keywords: *Backpropagation, Merged Convolutional Neural and Subsampling Techniques Layers, Convolutional Neural Network, Gender Classification*

1. Introduction

The question of gender categorization first surfaced in psychophysical research, which aimed to comprehend vision processing and discover distinctive traits that are used to discriminate between male and female subjects [1]. Additional investigations have demonstrated that it is possible to use the differences in facial masculinity and femininity to improve the functionality of face recognition systems in several types of fields, such as computer vision, biometrics, human-computer interfaces, and surveillance. However, real-world circumstances present a significant problem in managing the effects of lighting, position changes, facial emotions, occlusions, backdrop distortions, and noise on facial pictures. Therefore, to achieve high, reasonable, and accurate classification performance, these problems must be addressed in creating a substantial gender categorization system reliant on face analysis.

The conventional method of face recognition, which includes gender categorization as well, involves the sequential steps of feature extraction, dimensionality reduction, image processing, and classification. An efficient feature extractor requires previous application domain knowledge to build. It might be difficult to ascertain the optimal mix of classifiers to achieve high classification accuracy since it depends on the method employed for feature extraction. Furthermore, alterations to the issue domain may require a thorough system restructuring. A sequential procedure comprising picture capture and processing, dimensionality reduction, feature extraction, and classification is used in the classic method of face recognition, which includes gender categorization. A prior understanding of the application area is required to create an efficient feature extractor. Selecting the right classifier is essential since it affects the feature extraction technique used, and it can be difficult to identify which combination would yield the best classification accuracy. Furthermore, alterations to the issue domain frequently necessitate a total system overhaul.

A type of neural network that combines convolutional, subsampling, and densely connected layers is called a convolutional neural network, also known as CNN. Compared to conventional methods, this network structure which is shown in Figure 1 offers several advantages in pattern recognition. Within a single network, the CNN can do classification, feature extraction, and dimensionality reduction all at once. This integrated strategy maintains efficiency and cost-effectiveness while improving recognition accuracy.

Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India

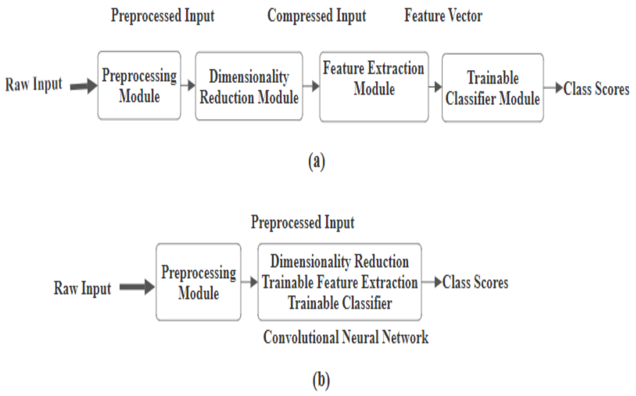


Fig 1. Pattern recognition methodologies: (a) Traditional, (b) CNN-centric

One effective method that integrates the integration of feature extraction and classification within a unified network structure is the convolutional neural network (CNN). In contrast to conventional techniques, CNN gains the capability to derive pertinent features by modifying feature abstraction weights during training. This makes human feature selection unnecessary and enables the framework to adjust to various input data types without requiring a lot of preprocessing. Furthermore, despite maintaining the spatial configuration of the input data, the CNN shows some invariance to geometric distortions, transformations, and modifications in 2D shape. Because face photos can differ in position, expression, lighting, and occlusion, CNN is well-suited for applications like gender categorization. CNN's ability to specifically address the drawbacks of conventional feature extractors is one of its fundamental advantages. CNNs are crucial to the training process because they are dynamically created and tightly coupled with the trainable classifier, in contrast to their static counterparts [3]. In conclusion, CNNs are more manageable to train than fully connected multilayer perceptron (MLP) neural networks with equivalent hidden layers but fewer parameters. This property has resulted in notable achievements encompassing a wide variety of applications, including face identification [5], character identification [4], person tracking [6], and tasks like traffic sign identification, among others.

This research presents a fresh perspective on gender categorization instantaneously employing a convolutional neural network (CNN). Here is a summary of this study's main contributions. To begin with, we provide a simple and effective 4-layer CNN designed for gender categorization based on face data in real time. This network architecture is characterised by a lower design complexity than previous approaches, with fewer layers, nodes, trainable coefficients, and connections. The filtering and down-subsampling layers are fused using a unique technique, and the convolution process is replaced with cross-correlation. Specifically, cross-correlation eliminates the need for weight flipping in the kernel matrix, which improves network performance.

2. Previous Work

Although gender categorization is important in numerous computer vision applications, there is a lack of study of recognition and identification, which are more common problems. The field of gender categorization has not produced as much research as these well-explored domains, where solutions often include using heuristic-driven feature extractors simultaneously with trainable or non-trainable classifiers. The short summary of previous efforts in this section underscores the many categorization techniques that have been used to tackle this particular problem. Among the algorithms often used for classification problems, the support vector machine (SVM) is notable. Regarding gender classification, a noteworthy contribution is reported in [7], where a system employing a polynomial kernel SVM and local binary pattern (LBP) obtained an impressive classification success rate of 94.08% on the CAS-PEAL face database. With a 3.0 GHz CPU and MATLAB 6.1, the implementation showed an average computation time of 0.12 s. A significant disadvantage of this methodology is that choosing the block magnitude for the LBP operator is a complex process that must be completed to get good classification results. A noteworthy undertaking, expounded upon in reference [8], demonstrated a remarkable 99.30% classification accuracy on the SUMS facial database. This technique included the K-means closest neighbour (KNN) classifier, Viola and Jones face identification, and 2D-DCT feature extraction. 2D-DCT is a computationally demanding method that is not appropriate for real-time applications, even with its great accuracy.

The first neural network-based gender categorization system was presented by [9], which combined many image processing modules with a tightly integrated Multilayer Perceptron (MLP) to produce an average discrepancy rate of 8.1%. Nonetheless, when considering current outcomes, this error rate is rather large. A hybrid strategy suggested in [10] employed principal component analysis (PCA) for facial picture dimensionality reduction, followed by the implementation of a genetic algorithm (GA) to choose a subset of eigenattributes. This approach is limited by the high computing complexity of the GA, even if it achieves an average inaccuracy of 11.30%. A prevalent constraint shared by all approaches is the division of feature retrieval and categorization modules, which frequently demands previous domain-specific expertise for ideal preprocessing and characteristic extraction designs.

3. Some Foundational Context of The Convolutional Neural Network (CNN)

A particular kind of forward-propagation network framework known as a convolutional neural network (CNN) is defined by the addition of several layers consisting of filtering and down-subsampling filters, followed by densely connected layers. Conventional CNN designs are modelled after the classical LeNet-5 CNN, which is seen in Figure 2. This framework was initially outlined by LeCun et al. in [4], and it became well-known for its effective use in hand-scripted digit detection tasks. Remarkably, LeNet-5 functions on visuals with a resolution of 32 × 32 pixels and consists of six processing layers (not including the input layer).

Three convolutional layers, identified as C1, C3, and C5, comprise the operational layers of the convolutional neural network (CNN), as seen in Figure 2. Two pooling layers, S2 and S4, are sandwiched between these convolutional layers to create a concluding stage that is designated as F6. Feature maps are the planes into which the feature extraction and reduction layers are organised. Individual neurons in each convolutional layer are locally coupled to tiny input areas in the previous layer that are 5 by 5 in size, called the receptive field [1]. Notably, neurons in the equivalent feature map create what is called a kernel because they exhibit a shared set of weights. This technique, known as local weight sharing, uses all feature maps that are produced when the kernel is convoluted with each corresponding receptive field; however, various feature maps within the same layer use different kernels.

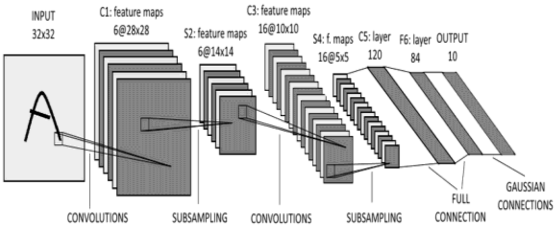


Fig 2. The configuration of the original LeNet-5 CNN

Spatial downsampling of feature maps occurs inside subsampling layers, reducing the map dimensions by a factor of two. The distinctive map in layer C1, for example, which is originally 28 × 28, is down-subsampled to generate a 14 × 14 feature map in layer S2, which comes next. The outcome layer F6, which is organised as a Multilayer Perceptron (MLP), completes the final classification process. The CNN design is fundamentally established on the architectural ideas of weight sharing, spatial subsampling, and regional perceptive fields [2, 6].

To facilitate feature extraction through convolution with these shared kernels, weight sharing entails using the same input sets of weights (also known as kernel weights) for every neuron in a feature map [3, 11]. Thus, a feature map inside a CNN is a two-dimensional representation of features that have been obtained from the input picture or feature map(s). This configuration, typified by spatially arranged neurons with spatially symmetric kernel weights, is reminiscent of the structural design of biological visual systems.

There are several benefits to this design approach. It first efficiently lowers data dimensionality and network complexity [1, 11]. In addition to helping with feature extraction, weight sharing may be used instead of weight removal, which helps make effective use of the machine's memory. To reduce dimensionality in the CNN, a down-subsample is used to preserve relevant data in the input plane and lessen susceptibility to changes and distortions in the detected characteristics [3]. Further information is given in [4] for a thorough knowledge of the LeNet-5 CNN algorithms.

4. CNN Proposal for Gender Classification

We outline the details of the CNN that was developed with automatic gender recognition in mind in this section. A thorough narration of the network architecture is included, coupled with the relevant formulas and algorithms that support its operation. We also discuss the training algorithm that we used to create our CNN and explain the procedural challenges connected with its optimisation and improvement.

4.1. Architecture

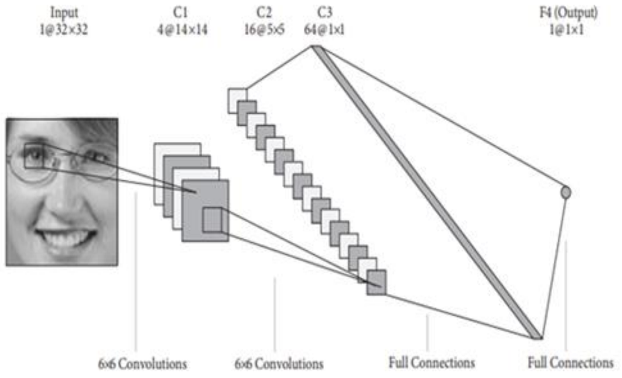


Fig 3. Recommended CNN design for gender recognition.

An outline of the network framework built into our CNN may be found in Figure 3. Four different processing layers are included in this architecture: an output layer called F4 and three feature extraction layers called C1, C2, and C3. A 32 × 32 pixel 2D face picture can be accommodated in the first input layer. Following the input layer, the convolutional layers C1, C2, and C3 are identified by four 14 × 14, sixteen 5 × 5, and sixty-four 1 × 1 feature maps, in that order. The output layer, which is Layer F4, has a single perceptron, commonly known as a 1 @ 1 x 1 map. This organized configuration outlines the fundamental architectural components coordinating the activities inside our CNN.

4.2. Formulation: Combination of Convolutional and Subsampling Layers

Using a fusion approach, the proposed convolutional neural network (CNN) maintains or improves picture categorization and identification performance while reducing operations to optimise the network design. This fusion methodology is built upon a methodology that was initially introduced by Simard et al. [12] and subsequently improved upon by Mamalet and Garcia [13]. This technique minimises the number of layers by merging a reduction layer and a convolutional stage into a single entity. The sequential convolutional and subsampling layers are swapped out for an individual convolutional layer with strides of two in this approach.

Our implementation of this approach has been exceedingly effective in reducing the entire number of layers in the CNN while maintaining or improving its performance. The CNN architecture involves a feature map, yj(l)(x,y) in layer 1, which can be symbolized by the following equation:

$$Y_j^{(l)}(x,y) = f\left(\sum_{i=0}^{N}\sum_{u=0}^{K_x^{(l)}}\sum_{v=0}^{K_y^{(l)}} Y_i^{(l-1)}\left(S_x^{(l)}x + u, S_y^{(l)}y + v\right) w_{ji}^{(l)}(u,v) + \theta_j^{(l)}\right)$$

Where $Y_i^{(l-i)}$ and $Y_j^{(l)}$ are input and output characteristic maps respectively,

f() denotes the activation function,

$w_{ji}^{(l)}$ represents the convolutional kernel weights

$\theta j^{(l)}$ is the bias,

N is the total number of input feature maps,

$S_x^{(l)}$ is the horizontal convolutional step size,

$S_y^{(l)}$ is the vertical convolutional step size, and

$K_x^{(l)}$ and $K_y^{(l)}$ represent the width and height of convolutional kernels.

The resulting output dimensions W(l) and H(l) of the feature map can be calculated using the formulas :

$W(l) = W(l-1)-K_x^{(l)}.S_x^{(l)}+1$,

$H(l) = H(l-1)-K_y^{(l)}.S_y^{(l)}+1$,

Where W(l-1) and H(l-1) denote the breadth and height of the input characteristic map respectively.

In summary, the fusion process that we have integrated into our CNN design proves to be an extremely effective strategy that combines the convolutional and subsampling layers into one cohesive unit.

Furthermore, this integration improves overall network design, but it also simplifies operations. Furthermore, this technique illustrates that it is possible to minimize the overall total layers in our CNN and yet maintain or improve its performance on tasks like image identification or classification. Note that cross-correlation is used, substituting convolution with the previously discussed process. It is important to recognize that convolution and cross-correlation are similar processes in image processing. The main difference is that convolution requires both side-to-side and up-and-down flipping of kernel weights. It harmonizes with our overarching aim to advance the effectiveness and efficiency of our CNN architecture to apply cross-correlation. The common expression for a 2D discrete operation in graphics processing should be taken into consideration for a more thorough understanding.

$$Y(x,y) = \sum_{u=0}^{X}\sum_{v=0}^{y} X(x-u, y-v)w(u,v)$$

$$Y(x,y) = \sum_{u=0}^{x}\sum_{v=0}^{y} X(x+u, y+v)w(u,v)$$

It is feasible to differentiate Equation (1) from Equation (2) primarily because the latter does not include the inversion of kernel weights. With the aid of the figures in Figure 6, these computational activities are made visually clear. Figure 6a displays a convolution kernel example. As is noticeable in Figure 4b, the traditional method creates a 2D discrete convolution by reversing the kernel weights both vertically and horizontally over an overlapping input plane. In contrast, the identical procedure is demonstrated in Figure 4c, with the obvious difference being that the kernel is not folded. It's interesting to observe that flipping has little consequence on the convolutional layer's initialization of values since the convolution kernel's values are established at random.

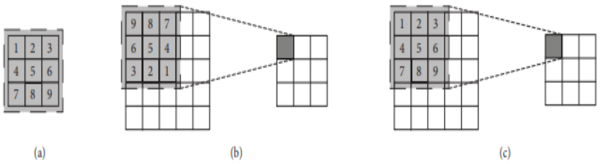


Fig 4. The two-dimensional discrete convolution operation:

(a) a convolution kernel example, (b)Convolution with flipped kernel weights, and (c) kernel weight convolution without flipping.

Both advancing and retreating propagations experience processing time increases due to flipping processes. Hence, when working with sizeable kernel dimensions and numerous convolutions over a prolonged duration of training iterations

(epochs), it is preferable to modify and use cross-correlation rather than convolution.

In this study, we utilize in the convolutional layers a scaled hyperbolic tangent function, as given by the following equation:

$$f(x) = A \tan h(Bx)$$

where A is indicative of the function's amplitude and B describes the function's slopes at the origin. By the suggestion of LeCun et al. [4], A and B are set to 1.7159 and 2/3, respectively.

For the output layer F4, a solo perceptron is employed to deduce the classification of a given input pattern. The value within the F4 output neuron is represented by the following equation:

$$Y_j^{(l)} = f(\sum_{i=1}^{N} Y_i^{(l-i)} w_{ji}^{(l)} + \theta_j^{(l)})$$

Every neuron's activation function is denoted by f( ), the aggregate number of input neurons is N, the numerical value of each input neuron is Yi(l-1), the neural connection weight that connects the input and end result neurons is wji(l), and the bias is characterized by θj(l). For this pattern, the masculine gender is indicated by an output value of +1, and the feminine category is indicated by an outcome value of −1.

Layer	Number of operations
Convolutional layer Ci	$WiHiKi^2$
Subsampling layer Si	($WiHiKi^2$ +1) / 4
Fused layer CSi	$(4(K_i + 1)^2 + (K_{i+1} +1)^2)$
Speedup factor,$(Ci, S)/CSi$	$(16K^2_i + 4K^2_i+1) / 4(K_i+1)^2 +(K_{i+1} +1)^2$

Table 1. Comparison of backpropagation algorithm complexity between the integrated convolutional and subsampling layer structure and the conventional CNN architecture. [13].

The backpropagation training technique in the traditional CNN architecture and the CNN with combined convolutional and subsampling layers are compared in Table 1 [10]. As can be seen, the fused convolutional layer outperforms the convolutional layer and sub-sampling layer combo employed in the traditional CNN design in terms of computing speed.

We find that when weight flipping is not there, the classification performance improves, and speed benefits from a merged convolutional layer.

This is covered in greater detail later in Section 4. Contrary to what they claim in their papers, the academicians, for instance, in [14] and [15], employed cross-correlation rather than convolution in the convolutional layers. It ought to be made apparent.

4.3. Training algorithm

The most widely adopted neural network learning technique is backpropagation. We compute error gradients, move them from the final layer towards the initial layer, and adjust weight values to reduce network error. However, one of the fundamental disadvantages of traditional backpropagation is its extremely sluggish convergence rate. Second-order techniques dynamically adjust the learning speed parameters to speed up network convergence.

This study uses the probabilistic diagonal Levenberg-Marquardt.(SDLM) algorithm, a second-order backpropagation technique that was initially presented by LeCun et al. in [16]. An annealed global. A learning rate is added to the process to enhance it and speed it up network convergence. The method that follows illustrates CNN's training procedure. As an additional learning mechanism, backpropagation is used in addition to the SDLM method. To initialize the kernel weights, Initialize Weight() generates them uniformly at random from a distribution [0.05, 0.05]. The superior outcomes are obtained with a uniform distribution, as shown by experiments with weight initialization techniques and parameter values.

Algorithm. Training procedure of CNN.

```
InitilazeWeight (CNN);
while not reaching convergence do
    CalculateLearningRateSDLM(sample);
    Shuffle(samples);
    for each training sample do
        output ← ForwardPassCNN(samples);
        loss ← CalculateError(output);
        error←BackwardPassCNN(loss);
        WeightUpdate(error);
    end for
end while
```

The Hessian matrix is roughly represented as the Jacobian square of the error of the network output concerning the weights using the SDLM method. The approximation provides a quicker convergence rate, guarantees invertibility, and produces appropriate computing costs [16]. Annealed global learning rates are used in the Calculate Learning Rate SDLM (samples) method to refine the SDLM learning algorithm. The following method may be employed to calculate the variable learning rate for each unique weight (or bias).

$$n_{ji} = \frac{\in}{\frac{d^2y}{dw_{ji}^2} + U}$$

where the regularization parameter is µji, the fine-tuned global learning rate is ϵ, and the weighted convergence factor is wji. The acceleration of change in the output approximated concerning weight is found by employing an equation:

$$\left(\frac{d^2y}{dw_{ji}^2}\right) new = (1-\rho)\left(\frac{d^2y}{dw_{ji}^2}\right) old + \rho\left(\frac{d^2y}{dw_{ji}^2}\right) current$$

where the consequence of the previous value is employed to calculate the future learning rate parameter, and γ is a minor memory constant. The following is the annealed global learning rate:

$$\in^{t+1} = \begin{cases} \in_{max} \; t = 0 \\ \in_{min} \; \in^t < \in_{min} \\ \in^t \times \alpha \quad otherwise \end{cases}$$

where σ min denotes the lowest learning rate on a global scale, σ max the initial learning rate on a global scale, and α the global learning rate's fading factor. The values of ε = 0.03, γ = 0.00001, ϵmax = 0.001, ϵmin = 0.00001, and α = 0.8 are employed in the computations of the learning rate. The processing is executed in a singular instance every two training epochs. The matching learning rate is modified for each unique weight or bias. The training procedure's seventh step locates the network fault. The mean squared error (MSE), our loss function, has the following mathematical expression:

$$L_P = ½ (Y_p – D_p)^2$$

where D_p denotes the intended output value for a given pattern P, Lp represents the outcome of the loss function or the mean square error, and Yp indicates the authentic consequence value.

In step nine, adjustments are made to the kernel's weights during the training process. The weight adjustment procedure in this instance is similar to the standard backpropagation in that it uses the following mathematical relations to update each weight or bias with its associated learning rate:

$$W_{ji}(t+1) = W_{ji}(t) - \eta_{ji}\frac{\partial Y^P}{\partial w_{ji}(t)}$$

$$\theta_j(t+1) = \theta_j(t) - \eta_j\frac{\partial Y^P}{\partial \theta_j(t)}$$

where, for the subsequent training iteration, wji (t + 1) and θj (t + 1) represent the updated weight and bias values.

5. CNN Framework Proposed for Gender Identification

The outcomes of our experiment on CNN-based gender categorization are shown in this section. The suggested CNN algorithm was written in C and built as a single-threaded, optimization level 3 (O3) software with GCC on Ubuntu 12.04 LTS. Our CNN is driven by an Intel Core 2 Duo T8300 (2.4 GHz) processor with three gigabytes of RAM.

The program is cross-compiled and executed on the Stratix III FPGA system-on-chip platform, tailored for utilization on an Altera Nois II CPU featuring 1 GB of RAM and a clock speed of 280 MHz

5.1. Databases and Arrangement of Dataset

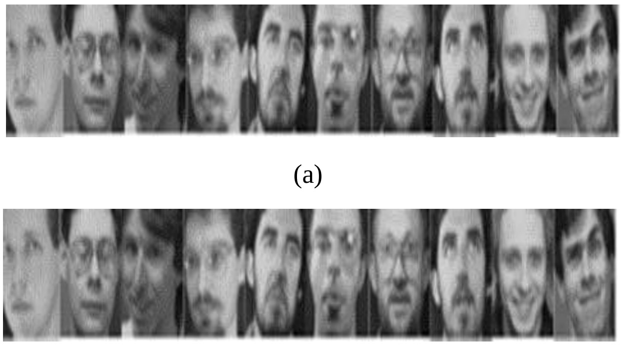
The trained CNN's classification performance is assessed using the AT&T and SUMS facial datasets, both accessible to the general public. We use our gender categorization technique to evaluate performance using the SUMS face database and benchmarking. There are 400 faces in the Stanford Medical Student Face Database or SUMS. 200 × 200 grayscale pictures for every topic. There are 200 subjects in the database: 200 males and 200 females. There is minimal to no difference in the frontal and upright lighting of the figures. The subjects exhibit diverse facial expressions and characteristics, whether or not they wear glasses.

The initial picture size of each of the 10 people (36 males and 4 females) in the AT&T facial database, originally recognized as the ORL database, was 92 × 112 pixels. Subjects were photographed with their facial expressions turned towards the front and slightly slanted to allow for some side movement against consistently dark backgrounds with different illumination. There are differences in face photographs in terms of facial characteristics like whether or not spectacles are worn and, in addition, expressions like open or closed eyes and smiling. The primary characteristics of the face photos in the SUMS face database are eyes, eyebrows, nose, mouth, beard, and spectacles if any are carefully edited out of the photographs. Subsequently, a size reduction is implemented to 32 by 32 pixels. Before being downsized to 32 by 32 pixels, the raw face photos from the AT&T facial images in the database were subject to the manual cropping window size of 92 × 92 pixels. Figures 4 and 5 display cropped picture samples for the SUMS and AT&T datasets, respectively.

(a)

(b)

Fig 5. Depicted are facial images from the SUMS database after cropping, highlighting :(a) male participants and (b) female participants.



(a)

(b)

Fig 6. Images of faces cropped in the AT&T database: (a) male participants, (b) female participants.

The processed face images undergo local contrast for image improvement and normalization techniques. The pixel values are normalized to lie between -1 and +1, leveraging the ensuing formula:

$$x = (x - x)\left(\frac{max - min}{x_{max} - x_{min}}\right) + min$$

The technique of local contrast normalization is applied and executed on the cropped facial images. Implementing the following formula, the pixel values are normalized to fall between -1 and +1:

Table 2 summarizes the findings of randomly dividing the preprocessed (normalized and cropped) photos into training and testing sets. The training and testing sets do not contain any visuals of the equivalent subject.

Face database	Training			Testing			Total
	Male	Female	%	Male	Female	&	
SUMS	120	120	60	80	80	40	400
AT & T	216	24	60	144	16	40	400

Table 2: Split of face databases into training and testing sets.

5.2. Findings regarding categorization accuracy

The outputs embedded in the network are shown in Figure 7 as an example of training pictures. By using convolutions in layer C1, three distinct complex properties are derived from the input picture. This involves convolving these features with 36 trained kernels to get 16 more basic 5 x 5 features. Subsequently, a sequence of pixels generated through the evaluation of these basic features represents the conclusions of a fully connected layer. The conclusion neuron facilitates the final classification. Depending on its color, a pixel within this examination, a value of +1 or -1 is assigned.

The input configuration is identified and labelled as male if Y^P ≥ threshold and as female otherwise in terms of gender recognition. The criterion for this study is 0. After the network is exposed to every input pattern in the dataset, the number of accurate classifications may be found by counting the right classifications. The classification rate is subsequently calculated using the following formula:

$$Classification\ Rate = \frac{Correct\ Classifications}{Total\ testing\ samples} \times 100\%$$

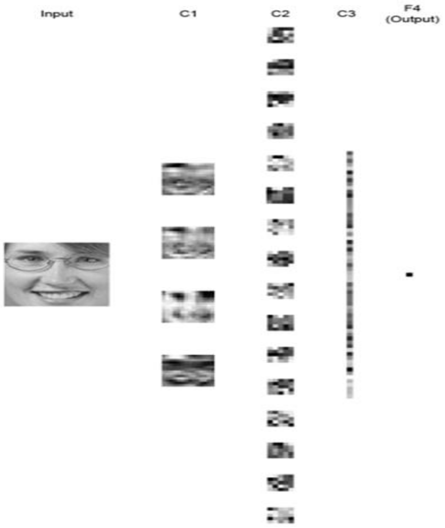


Fig 7. Feature maps for a training image across diverse CNN layers.

For the AT&T and SUMS datasets, the testing and training mistake patterns are shown in Figure 7, respectively. The initial few epochs see larger mean squared discrepancies (MSEs) during training than testing, but after around six epochs, they start to decline. The content of the training dataset is bigger than outperforming the testing set, which is one reason for this discrepancy.

Training and testing errors go down as training grows until they attain a stable number, indicating the network's convergence. As the MS database scenario with its high gradient in Figure 7 illustrates, it is imperative to implement the second-order strategy for network learning, namely SDLM.

By not exceeding twenty training cycles, as seen by the misclassification error progression over epochs graph in Figure 8, our CNN obtains convergence. We obtained a great score. Obtaining a classification accuracy of 98.75% on the SUM database and an amazing classification rate of 99.38% for the AT&T Face database. Among the examination of 160 testing instances in the AT&T dataset, there is just one mistake, making it exceptional.

Mean Squared Errors (MSE) for Training and Testing

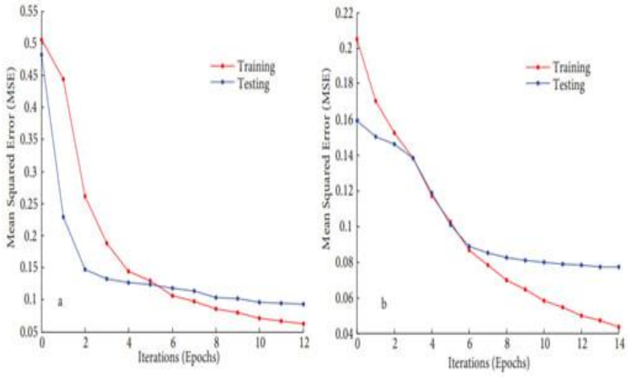


Fig 8. Visualization of error fluctuations in

(a) SUMS and (b) AT&T databases across training and testing

5.3. An analysis of how weight flipping affects classification accuracy

Unlike the traditional 2D discrete convolution, we introduce the CNN architecture introduced in this investigation, which executes cross-correlation operations within the fused convolutional layer. The AT&T and SUMS face datasets are utilised to examine the consequences of weight flipping in convolution kernels on classification performance. According to the data presented in Table 3, the experiment was carried out by applying the constant learning rate strategy and parameter values in both cases.

Our results showed a substantial rise in the classification rate (1.87%) for the SUMMS dataset. Similar findings are obtained from an examination of the AT&T database, where classification accuracy increased from 97.50% to 99.38%. Our results show that using cross-correlation (without flipping the weights) instead of convolution (with the kernel weights flipped) improves the classification performance of our CNN.

Misclassification Errors of Training and Testing Set

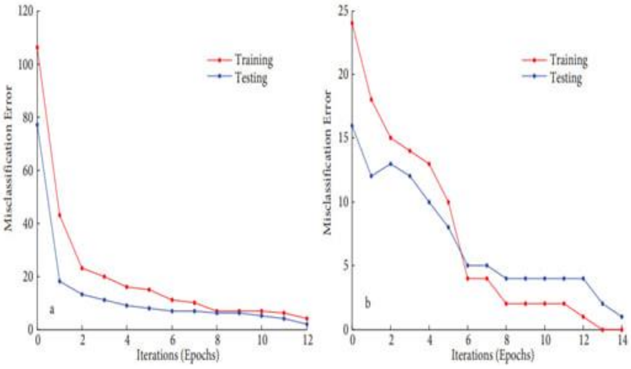


Fig 9. Mistakes in the misclassification of the

(a) SUMS database and (b) AT&T database

CNN convolutional layer applying:	Classification rate(%)	
	SUMS	AT &T
Convolution(i.e kernel weights are flipped)	96.88	97.50
Cross-correlation(i.e kernel weights are not (flipped)	98.75	99.38

Table 3. CNN categorization rates are impacted by weight flipping.

5.4. Comparing Classification performance outcomes through benchmarking

The outcomes of our comparative analysis between our suggested CNN and other state-of-the-art research assessed on the AT&T and SUMS datasets are presented in Tables 4 and 5, respectively. Only the classification rate of our cited articles, which combined the application of DCT with a Kohonen self-organising map network, is worse than ours for the SUM dataset. This can be clarified by the observation that we decided to preserve the impact of light variations by not processing the cropped facial photographs.

Researcher(s)	Year	Method(s)	Classification rate(%)
Khan et al. [17]	2013	Decision tree	98.50
Majid et al. [18]	2003	DCT + Modified KNN	98.54
Nazir et al. [8]	2010	DCT + KNN	99.30
Proposed work	2014	CVN with fused convolution and subsampling layers	98.75

Table 4. SUMS face database-based benchmarking results.

Table 5 lists the research that classified people's genders using the AT&T face database. As far as we were aware, among all the strategies used in this study, the strategy we have provided delivers the greatest rate of classification. This demonstrates how well CNN performs in contrast to other feature extraction and classification techniques.

Researcher(s)	Year	Method(s)	Classification rate(%)
Jiao et al. [19]	2012	PCA + extreme learning machine	93.75
Basha et al. [20]	2012	Continuous wavelet transform & SVM	98.00
berbar [2]	2013	DCT + SVM	98.93
Proposed work	2014	CVN with fused convolution and subsampling layers	99.38

Table 5. The outcome of benchmarking using the AT&T Face Database.

5.5. Analysis of training accomplishments

Effectiveness in the training phase of the proposed method is compared with LeNet-5, a sample relative to conventional CNN architectures. Table 6 provides trainable parameter quantity linkages for each of the two situations. Our CNN exhibits a low overhead of 1.5% and a somewhat larger set of trainable parameters in comparison to the classic architecture. Having said that, the proposed CNN features connections 2.3 times higher than standard CNN. While an increase in learnable parameters requires more memory space, an elevated number of connections produces a greater computational load that could slow down the network.

CNN architecture	Total trainable parameters	Total connections
LeNet-5	26789	207,745
Proposed CNN	27189	88,997

Table 6. Enumeration of contrasts between two distinct architectures.

We assess the efficacy of each CNN design by examining the median processing duration needed. The average processing times are calculated for testing one pattern, 10 training epochs, and one iteration during the training epoch set at 240 and 480 patterns, respectively. The time required to initialise weight and read input pictures from memory is not taken into consideration in this evaluation.

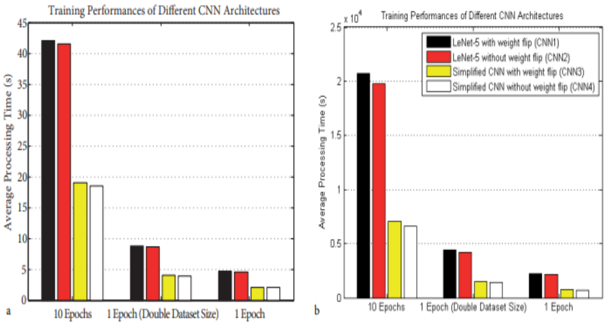


Fig 10. Training outcomes across several CNN architectures on

the (a) PC and (b) FPGA platforms.

Figure 10 shows performance analysis: LeNet-5 vs. our CNN processing times, including and excluding convolutional weight flipping. It's interesting to see that our suggested CNN shows the fastest processing time. We see that although CNN1 (LeNet-5 with weight flipping in kernels) requires around 4.65 seconds to complete a training epoch, CNN4 (our CNN without weight flipping) completes it in just 1.98 seconds. Probably, weight flipping will not have a substantial effect on the network's slowdown in one comprehensive training pass epoch, even though CNN3 our CNN that incorporates weight flipping achieves an epoch training time of 2.02 seconds. However, the discrepancies become more noticeable as training epochs are extended and datasets are bigger.

Over ten training epochs, there was a 0.64-second difference in the duration of processing CNN3 and CNN4. Similar trends have been observed on the FPGA platform. Notable is the facial database training, which is finished in under 15 epochs and takes less than a minute.

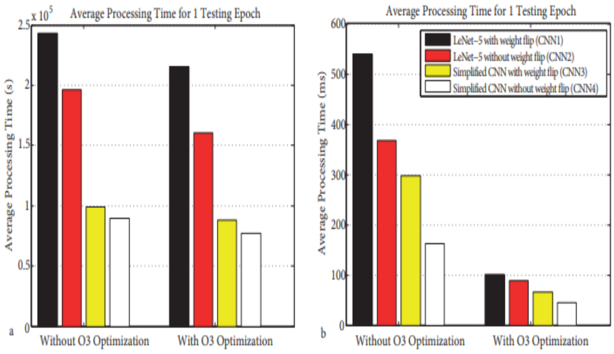


Fig 11. Average processing time for a testing epoch on a personal computer and an FPGA platform.

Figure 11 demonstrates a notable performance increase on a PC platform over an FPGA platform, most likely resulting from the latter's constrained hardware capabilities.

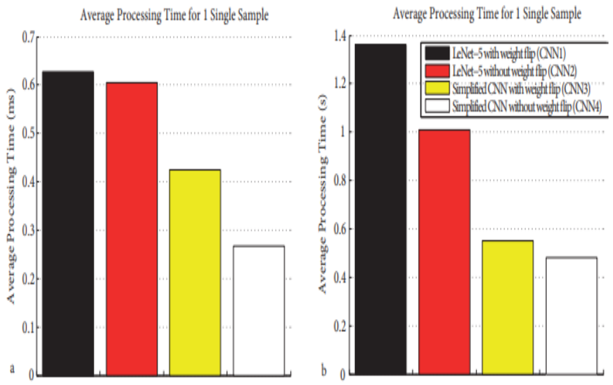


Fig 12. Average time consumption for processing one facial image on (a) personal computer, (b) FPGA system.

6. Conclusion

In summary, this paper provides an overview of a simplified network structure dedicated to gender categorization from face photos and a gender recognition system utilising ConvNets for real-time processing. The novel dimension of the model is the use of creative layer fusion and the replacement of conventional convolution with cross-correlation, which leads to a notable decrease in computing burden and design complexity. The CNN trained using fine-tuned global learning rates and second-order backpropagation achieves notable classification accuracy of 98.75% and 99.38%, respectively, when tested on both the SUMS and AT&T face datasets.

CNN's remarkable performance is further demonstrated by its remarkable throughput of exceeding 3700 photos per second, which it achieved by processing 32 x 32-pixel facial images not exceeding 0.27 milliseconds. The network converges quickly during training, using less than twenty epochs. Gender categorization is improved by the suggested CNN design, which uses both subsampling and convolutional layers and substitutes cross-correlation for convolution.

The benchmarking findings, which show 99.38% and 98.75% classification proportions on the AT&T and SUMS face datasets, respectively, validate the proposed CNN's improved classification skills. Specifically, the suggested architecture speeds up significantly—on a PC platform, it can evaluate and categorise 32 × 32-pixel input pictures in under 0.3 milliseconds. The present study is a groundbreaking investigation into the effects of weight fading in CNN systems, offering insightful information in the context of future research on problems like alignment and face identification.

7. Future Work

Investigating Face Recognition and Positioning: By broadening the scope of these outcomes to include tasks related to face identification and positioning with similar CNN architectures, it may be possible to formulate a system for gender recognition that is both comprehensive and efficient. Examining how well the optimised design performs in tasks other than classification may provide important new information about its wider applicability.

Maximising Hardware Utilisation: To make the suggested method more practically applicable, it should be customised for hardware implementation that is made especially for instantaneous processing in contexts with limited resources. Its efficiency might be further increased by investigating optimisations for execution on specialised hardware, such as Application-Specific Integrated Circuits (ASICs) or Field-Programmable Gate Arrays (FPGAs).

Strengthening and Broadening: To make the model more generalizable and resilient, it is essential to assess its performance in a variety of demographics, face expressions, and environmental contexts. A thorough examination in a range of scenarios would improve the model's capacity to adjust to real-world circumstances.

Handling Ethical Concerns and Mitigating Bias: Careful consideration must be given to ethical issues, especially those about bias detection and mitigation in gender identification systems. The model's predictions must be inclusive and fair to overcome biases and guarantee ethical deployment.

These next paths offer chances to enhance and improve the suggested CNN architecture's robustness, applicability, and ethical concerns. These developments may open the door for successful implementation in real-world situations, particularly in settings with limited resources.

References

[1] Tivive FHC, Bouzerdoum A. "A gender recognition system using shunting inhibitory convolutional neural networks." In: International Joint Conference on Neural Networks; 2006; Vancouver, Canada. New York, NY, USA: IEEE. pp.5336–5341.

[2] Khalajzadeh H, Mansouri M, Teshnehlab M. Face recognition using convolutional neural network and simple logistic classifier. Stud Comp Intell 2013; 223: 197–207.

[3] Strigl D, Kofler K, Podlipnig S. Performance and scalability of GPU-based convolutional neural networks. In: 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing; 17–19 February 2010;Pisa, Italy. New York, NY, USA: IEEE. pp. 317–324.

[4] LeCun Y, Bottou L, Bengio Y, Haffner. P. Gradient-based learning applied to document recognition. P IEEE 1998;86: 2278–2324.

[5] Duffner S. Face Image Analysis with Convolutional Neural Networks. Munich, Germany: GRIN Verlag, 2009.

[6] Fan J, Wei X, Ying W, Yihong G. Human tracking using convolutional neural networks. IEEE T Neural Networ 2010; 21: 1610–1623.

[7] Lian HC, Lu BL. Multi-view gender classification using local binary patterns and support vector machines. Lect Notes Comput Sc 2006; 3972: 202–209.1263

[8] Nazir M, Ishtiaq M, Batool A, Jaffar MA, Mirza AM. Feature selection for efficient gender classification. In: Proceedings of the 11th WSEAS International Conference; 2010. pp. 70–75.

[9] Golomb BA, Lawrence DT, Sejnowski TJ. SEXNET: A neural network identifies sex from human faces. In: Proceedings of NIPS; 1990. pp. 572–579.

[10] Sun Z, Yuan X, Bebis G, Louis SJ. Neural-network-based gender classification using genetic search for eigenfeature selection. In: Proceedings of the 2002 International Joint Conference on Neural Networks; 12–17 May 2002; Honolulu, HI, USA. New York, NY, USA: IEEE. pp. 2433–2438.

[11] Dawwd SA, Mahmood BS. Video Based Face Recognition Using Convolutional Neural Network. Rijeka, Croatia: InTech, 2011.

[12] Simard PY, Steinkraus D, Platt JC. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. Redmond, WA, USA: Microsoft Research, 2003.

[13] Mamalet F, Garcia C. Simplifying ConvNets for fast learning. In: Villa A, Duch W, Erdi P, Masulli F, Palm G, editors. Artificial Neural Networks and Machine Learning – ICANN 2012. Vol. 7553. Berlin, Germany: Springer,2012, pp. 58–65.

[14] Ji S, Wei X, Ming Y, Kai Y. 3D convolutional neural networks for human action recognition. IEEE T Patten Anal 2013; 35: 221–231.

[15] Mamalet F, Roux S, Garcia C. Embedded facial image processing with convolutional neural networks. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems; 30 May–2 June 2010; Paris, France. New York, NY, USA: IEEE. pp. 261–264.

[16] LeCun Y, Bottou L, Orr G, M¨uller KR. Efficient BackProp. Lect Notes Comp Sc 2002; 1524: 9–50.