A Gender Classification in Facial Images Using Convolutional Neural Networks: A Comparative Models Analysis

Juan Camilo García Perez. [*]

[*] *Department of Electronic Engineering, Universidad Santo Tomás, Bogotá, Colombia (e-mail: { juancgarciap} @usantotomas.edu.co)*

Abstract:
This research explores the effectiveness of Convolutional Neural Networks (CNNs) in the context of gender classification using facial images. The study compares three distinct CNN architectures, through comprehensive experimentation and analysis, it's evaluated their performance, and provides insights into the challenges and opportunities of employing CNNs for gender classification tasks.

Keywords:
Convolutional Neural Networks (CNNs), Deep Learning, Neural Networks, Classification, Gender Classification.

1. INTRODUCTION

Gender classification from facial images has become a significant application in computer vision, with implications ranging from security systems to personalized user experiences. Convolutional Neural Networks (CNNs) have emerged as powerful tools for image classification tasks, demonstrating remarkable accuracy and efficiency.

The fundamental process of a CNN involves a sequence of transformative layers designed to automatically learn hierarchical representations from input data. The first layer, a Convolutional 2D layer, convolves the input image with a set of learnable filters. This operation captures local patterns and features, enabling the network to recognize complex structures within the images. Subsequently, a MaxPooling layer is employed to reduce spatial dimensions and retain the most salient features. This pooling process enhances computational efficiency and aids in creating translation-invariant representations.

Following the convolution and pooling stages, the network incorporates a Flatten layer, which converts the multi-dimensional output into a one-dimensional vector. This flattened representation is then fed into densely connected layers, allowing the network to learn intricate relationships between features. The final layer often employs an activation function, enabling the model to produce probability distributions over classes.

In the context of gender classification, the CNN learns discriminative features from facial images that distinguish between male and female subjects. This study aims to delve into the nuances of CNN architectures, comparing their performance in gender classification tasks and exploring strategies for enhancing accuracy through techniques like data augmentation.

2. OBJECTIVES

2.1 Main Objective

To evaluate the efficacy of Convolutional Neural Networks in gender classification using facial images.

2.2 Specific Objectives

- To compare the performance of three different CNN architectures in gender classification.
- To analyze the model's metrics to identify patterns of misclassifications in gender prediction.
- To evaluate the influence of hyperparameter tuning on the overall performance.

3. PROCEDURE

3.1 Dataset Preparation

The dataset consisting of labeled facial images representing both male and female subjects was used for training and testing the CNN models, the data set was obtained in kaggle it's called "Gender Detection and Classification - Image Dataset" [] which was verified in order to have the correct information on it and it's composed by the training folder of 220 images and the test folder of 80 images for the genders 'Male' and 'Female'.

3.2 Model Architectures

The code establishes a Convolutional Neural Network (CNN) for image classification using the Keras library. The model architecture includes a 2D convolutional layer with 32 filters and a ReLU activation function, followed by max-pooling for spatial downsampling. A Flatten layer converts the 2D output to a vector, connecting to two dense layers with 256 neurons each and ReLU activation. The final layer, employing the softmax and sigmoid activation functions, outputs two classes for binary classification.

The model is compiled with the Adam optimizer, utilizing sparse categorical crossentropy loss for integer labels and accuracy as the evaluation metric. Data augmentation is implemented during training through ImageDataGenerator, applying rescaling, shearing, zooming, and horizontal flipping. Training and testing datasets are loaded from directories, specifying target size, batch size, and sparse class mode.

The model is then trained over 100 epochs, utilizing the training set and validating on the testing set. Overall, the code encapsulates the construction, compilation, and training of a CNN tailored for binary image classification with a focus on women and men categories.

The models were trained on the prepared dataset, and the training process included augmentation techniques to enhance the variety of the training data. The training progress was monitored using validation data.

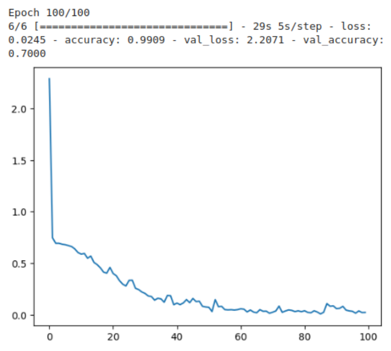
4. RESULTS

4.1 Architecture 1: Softmax Activation Function

For the first architecture was used the softmax activation function because is an extension of the sigmoid function and is commonly used in multi-class classification problems. It is defined as:

$$Softmax(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{1}$$

for each element i in the input vector x. The softmax function takes an input vector and normalizes it into a probability distribution over multiple classes. It ensures that the sum of the probabilities for all classes is equal to 1[3], making it suitable for problems where an input can belong to one of several mutually exclusive classes. Using this architecture was obtained the following loss metric.



```
Epoch 100/100
6/6 [==============================] - 29s 5s/step - loss:
0.0245 - accuracy: 0.9909 - val_loss: 2.2071 - val_accuracy:
0.7000
```

Fig. 1. Softmax Loss Error.

The model seems to have performed well during training, as evidenced by the low training loss (0.0245) and high training accuracy (99.09%). However, the validation loss (2.2071) and validation accuracy (70%) indicate that the model may not generalize well to unseen data. This discrepancy between the training and validation performance suggests a potential issue with overfitting, where the model has memorized the training data but struggles with new, unseen data.

Now, let's verify the confusion matrix performance (Fig 2). The confusion matrix further supports this interpretation. The diagonal elements ([21, 17]) represent correct predictions (true positives and true negatives), while the off-diagonal elements ([19, 23]) represent incorrect predictions (false positives and false negatives). The higher number of false positives and false negatives may indicates that the model is making some errors in classification, contributing to the lower validation accuracy.

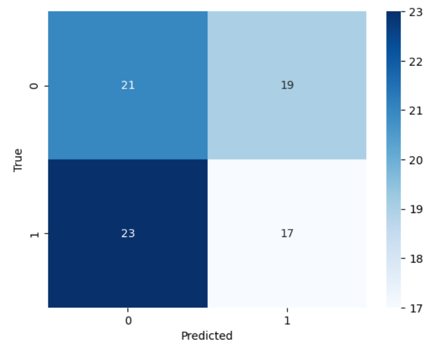
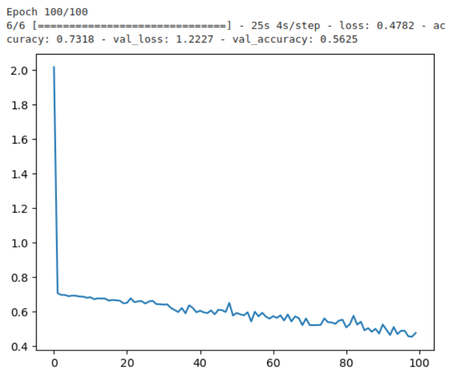


Fig. 2. Softmax Confusion Matrix.

4.2 Architecture 2: Softmax Activation Function with hyperparameters

For this architecture was decided to update the hyperparameters of the architecture shown before in order to enhance its performance, the modifications made were adding the following data augmentation configurations: $rotation\_range = 20$ for additional rotation, $width\_shift\_range = 0.2$ for additional width shift and $height\_shift\_range = 0.2$ for additional height shift. And it loss metric and confusion matrix obtained are Fig. 3 and Fig. 4. respectively.



```
Epoch 100/100
6/6 [==============================] - 25s 4s/step - loss: 0.4782 - ac
curacy: 0.7318 - val_loss: 1.2227 - val_accuracy: 0.5625
```

Fig. 3. Softmax Modified Loss error.

The training process appears to have resulted in a model with moderate performance. The training loss (0.4782) and accuracy (73.18%) indicate that the model has learned patterns from the training data, but there is room for improvement. The validation loss (1.2227) and accuracy (56.25%) suggest that the model struggles to generalize well to unseen data, indicating potential overfitting or suboptimal model architecture.

Analyzing the confusion matrix provides additional insights. The diagonal elements ([19, 24]) represent correct predictions (true positives and true negatives), while the

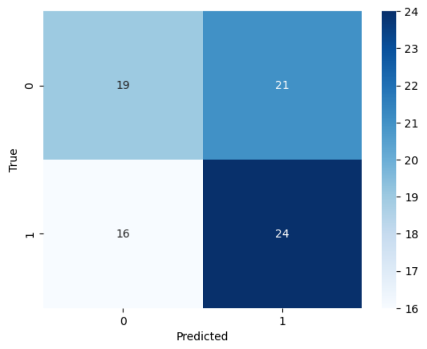


Fig. 4. Softmax Confusion Matrix.

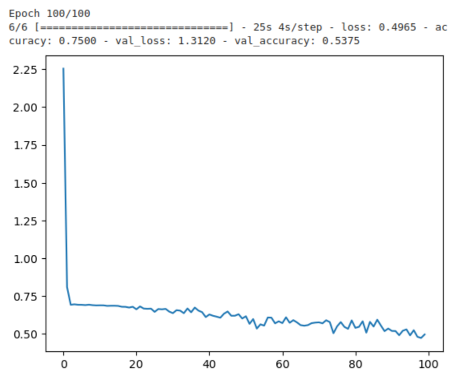
off-diagonal elements ([21, 16]) represent incorrect predictions (false positives and false negatives). The relatively balanced distribution of false positives and false negatives suggests that the model is making errors in both classes, indicating a need for further refinement.

4.3 Architecture 3: Sigmoid Activation Function

Observing the troubles that the systems were facing, implementing an architecture with more epoch it wont be a solution because it was already obtained signals that the architecture implemented before was already having an overfitting behavior. So modifying the activation function could be a good option to resolve this issue. The sigmoid activation function, also known as the logistic function, is defined as:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Where x is the input to the function. The sigmoid function squashes the input values between 0 and 1. It is particularly useful in binary classification problems where the goal is to predict a probability of belonging to one of two classes (0 or 1)[3]. The sigmoid function is applied independently to each neuron in the output layer of a binary classification model, providing a probability distribution.



```
Epoch 100/100
6/6 [==============================] - 25s 4s/step - loss: 0.4965 - ac
curacy: 0.7500 - val_loss: 1.3120 - val_accuracy: 0.5375
```

Fig. 5. Sigmoid Loss error.

The training results indicate that the model achieved a training accuracy of 75%, suggesting a decent ability to capture patterns in the training data. However, the validation accuracy of 53.75% suggests some difficulty in generalizing to new, unseen data. The relatively higher training accuracy compared to the validation accuracy may indicate a degree of overfitting, where the model memorizes the training set but struggles to perform well on new examples.

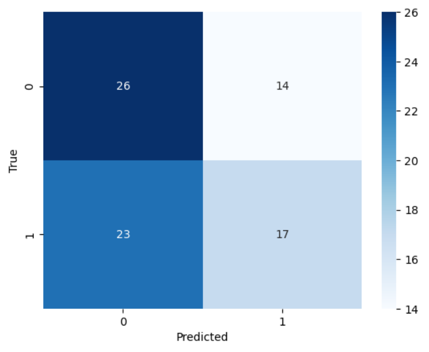


Fig. 6. Sigmoid Confusion Matrix.

Examining the confusion matrix provides further insights. The diagonal elements ([26, 17]) represent correct predictions (true positives and true negatives), while the off-diagonal elements ([14, 23]) represent incorrect predictions (false positives and false negatives). The imbalance in false positives and false negatives could indicate a bias towards one class or issues with model sensitivity and specificity.

To summarize the performance of each architecture observe the table 1, but in practice, what was the result by each architecture? The results are consigned in table 2.

Architecture	Epochs	Training Acc.	Validation Acc.	Confusion Matrix
1	100	99.09%	70.00%	[21, 19], [23, 17]
2	100	73.18%	56.25%	[19, 21], [16, 24]
3	100	75.00%	53.75%	[26, 14], [23, 17]

Table 1. CNN Architecture Comparison

Architecture	Correct Male	Correct Female	Correct Queer
1	1 correct	3 correct	0 correct
2	0 correct	3 correct	0 correct
3	3 correct	3 correct	0 correct

Table 2. Correct Classifications by Gender for Each CNN Architecture

The evaluation of three distinct convolutional neural network (CNN) architectures reveals varying degrees of success in classifying images based on gender categories. Architecture 3 emerges as the most robust, successfully identifying both male and female images. However, it is noteworthy that none of the architectures achieved correct classifications for queer images, it was a test in order to see how the system can identify other types of bodies, indicating potential challenges in recognizing this category. Architecture 1 demonstrated competence in identifying female images but struggled with the classification of males, while Architecture 2 faced difficulties in accurately classifying both male and female images. These findings underscore the need for ongoing refinement and optimization of CNN models, suggesting areas for improvement in the recognition of specific gender categories.

5. CONCLUSIONS

- Each architecture faced specific challenges in gender classification. While Architecture 1 excelled in recognizing females, it struggled with male classifications.

Conversely, Architecture 2 encountered difficulties in both male and female identifications.

- Across all architectures, recognizing queer images proved to be a challenging task, as none achieved correct classifications in this category. This highlights a potential limitation in the models' ability to capture the nuances of queer gender expressions.
- Architecture 3 emerged as the most robust, demonstrating success in identifying both male and female images. This suggests a superior ability to capture gender features and nuances compared to the other architectures.
- The influence of the number of training epochs on model performance is evident. The increased accuracy in later epochs indicates the importance of allowing models to train over an adequate number of iterations for improved classification.
- Discrepancies between training and validation accuracy suggest potential overfitting or generalization issues, emphasizing the importance of balancing model complexity and data size.
- The overall findings underscore the need for ongoing model refinement, with avenues for improvement identified in recognizing specific gender categories. Optimizing hyperparameters and employing advanced data augmentation techniques may contribute to enhanced model inclusivity and performance.

REFERENCES

[1] Gender Detection Classification - Image Dataset obtained: https://www.kaggle.com/datasets/trainingdatapro/gender-detection-and-classification-image-dataset/

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. URL: https://www.deeplearningbook.org/

[3] Keras contributors. Keras Documentation. URL: https://keras.io/

[4] Michael Nielsen. Neural Networks and Deep Learning. URL: http://neuralnetworksanddeeplearning.com/