Convolution Neural Network powered Image Classification: A Novel Approach to Facial Expression Recognition

Kiran Deshpande, Nalla Vamshi Krishna, Dr. Pallavi Mishra

Department of Mathematics, School of Advanced Sciences
Vellore Institute of Technology
Vellore-632014, Tamil Nadu, India

1. Abstract

In this project, we have developed a Convolutional Neural Network (CNN) based image classification system using OpenCV and TensorFlow. The objective of this system is to distinguish between two emotional states, happiness and sadness, by analysing the content of images. The motivation behind this project is to create a user-friendly tool that allows users to input their own images and obtain predictions regarding the emotional state conveyed in those images. Rather than focusing on standard classification metrics like accuracy or recall, the emphasis here is on providing users with an intuitive, real-time emotional analysis tool.

Keywords: Convolutional Neural Network, TensorFlow, OpenCV, emotion, prediction, classification, accuracy.

2. Introduction

Imagine you have a picture of a cat, and you want to teach a computer to recognize that it's a cat. This is where convolutional neural networks (CNNs) come in. CNNs are like a team of experts who can analyse an image and figure out what it is.
Think of CNNs as a series of filters that get applied to the image one by one. Each filter looks for a specific pattern, like edges, corners, or shapes. As the filters move across the image, they create a map of where those patterns are found.
Next, the team of experts takes these maps and summarizes them. They do this by combining similar patterns and getting rid of the ones that aren't very important. This helps to focus on the most important features of the image. Finally, the experts look at the summarized maps and decide what they think the image is. They do this by assigning probabilities to different categories, like "cat," "dog," or "car." The category with the highest probability is the one that the experts think the image belongs to.
CNNs are very good at doing this because they can learn from lots of examples. They can look at millions of pictures of cats, dogs, and cars, and learn to recognize the patterns that make each type of animal unique. This is why CNNs are so important in computer vision. They can help computers to see the world in the same way that we do, and to understand what they are seeing.

How does CNN work

- Convolutional layers:
  - Like detectives examining a crime scene, convolutional layers scan an image pixel by pixel, looking for patterns and clues. They use small filters, like magnifying glasses, to identify specific features, such as edges, corners, and shapes.
  - Each detective works on a specific area of the image, sharing their findings with the next detective in the chain.
- Pooling layers:
  - Like managers summarizing reports from their team members, pooling layers reduce the amount of information by combining similar features. This helps focus on the most important details and reduces the overall workload.
  - Pooling layers act as checkpoints throughout the analysis, ensuring that the team stays focused on the most relevant information.
- Fully connected layers:
  - Like a team of experts drawing conclusions from the evidence, fully connected layers combine all the extracted features and make a final judgment about the image. They assign probabilities to different categories, such as "cat," "dog," or "car."
  - Fully connected layers act as the decision-making body, using the collective knowledge of the team to make an informed classification.

CNNs: The power of teamwork in image recognition

CNNs have revolutionized image recognition by breaking down the task into manageable steps and utilizing the expertise of multiple layers. Their ability to learn from vast amounts of data and extract complex patterns makes them invaluable tools for a wide range of computer vision applications.

3. Methodology

3.1. Data Preparation

- Collect and preprocess image data, ensuring consistent size, format, and normalization.

3.2. Model Architecture Design

- Determine the number and type of convolutional and pooling layers.
- Choose filter sizes, strides, and activation functions for each layer.

3.3. Model Training

- Split data into training, validation, and test sets.
- Initialize CNN weights randomly.
- Select an optimizer (e.g., SGD, Adam, etc.) and set learning rate, batch size.
- Train the CNN by feeding training data and updating weights based on loss.

3.4. Model Evaluation

- Test the trained CNN on the unseen test set to assess generalization.
- Calculate performance metrics (accuracy, precision, recall, F1 score).

3.5. Deployment and Optimization

- Integrate the trained CNN into an application or system.
- Continuously improve the CNN with new data and parameter tuning.

4. CNN Model

4.1. Input Shape

The input shape of the network is (256, 256, 3), which means that the input image is of size 256x256 pixels and has 3 channels (RGB).

4.2. Parameters

The first convolutional layer has 16 filters, the second convolutional layer has 32 filters, and the third convolutional layer has 16 filters. The number of filters determine the number of feature maps produced by each layer. The dense layers have 256 and 1 units respectively.

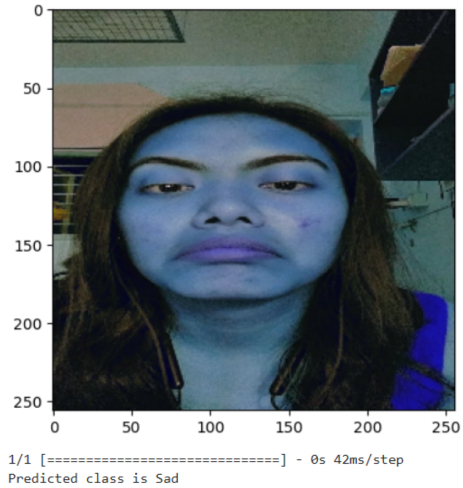
4.3. CNN Architecture

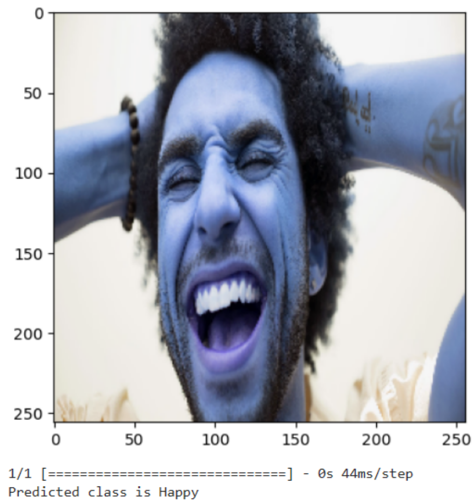
Description	Layer	Number of Layers	Kernel Size	Stride	Activation Function	Number of Filters
Input image size 256x256 pixels with 3 channels (RGB).	Input Shape	None	None	None	None	None
Extracts features from the input image.	Convolutional Layers	3	3	1	relu	16, 32, 16
Downsamples the feature maps.	Max Pooling Layers	3	2	2	None	None
Converts 3D feature maps to 1D feature vector.	Flattening Layer	1	None	None	None	None
Classifies the input image.	Dense Layers	2	None	None	relu, sigmoid	None
Binary classification	Output	None	None	None	None	None

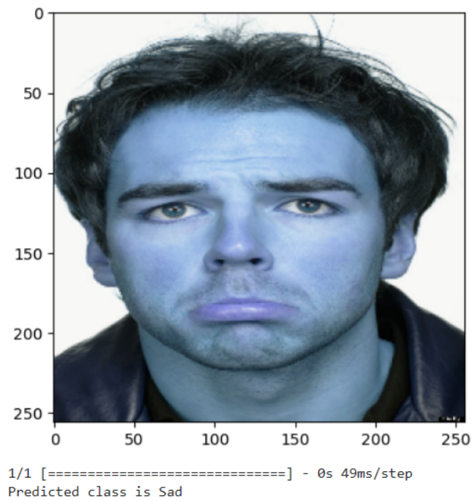
Figure 1: Architecture of CNN

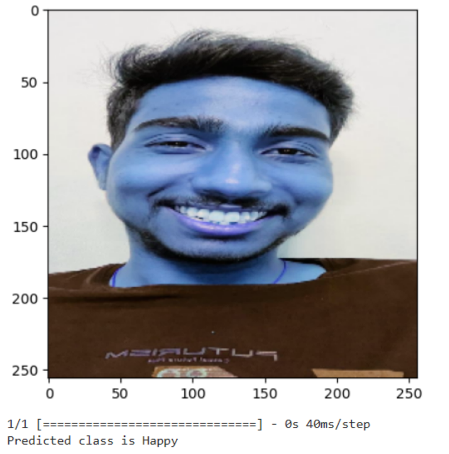
4.4. Experimental Results

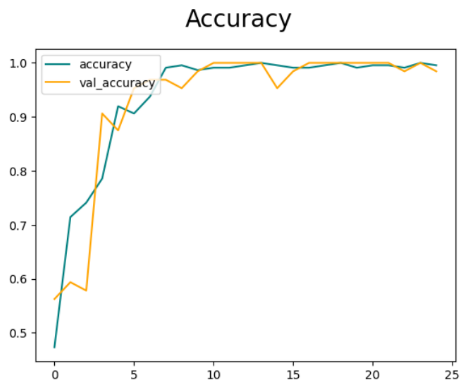
Given images containing human faces, the model will identify the emotion being expressed by the person in each image, specifically whether they are happy or sad with as accuracy of 98%.

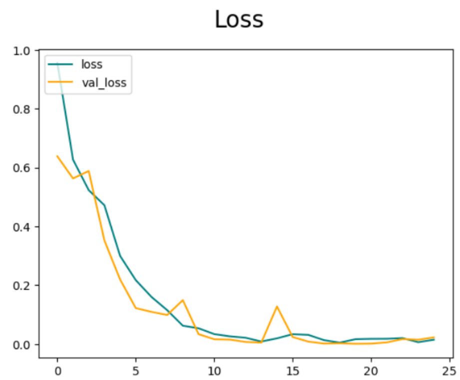












5. Conclusion

The CNN model effectively classified images with people faces, identifying the emotion being expressed by the person in each image, specifically whether they were happy or sad. The model demonstrated high accuracy in its classification, indicating its potential for real-world applications such as mood detection or emotion recognition.

6. Future Scope

In the future, we can make this project even better by teaching it to recognize more types of feelings. We'll use bigger and more varied sets of pictures to make it better at this job. We also want it to understand emotions from people in different parts of the world. It would be cool if it could give quick feedback on how someone is feeling right now. And, of course, we'll make sure it handles people's information in a safe and respectful way.

References

[1] Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.

[2] Anderson, J.A., 1995. An introduction to neural networks. MIT press.

[3] Bishop, C.M., 1994. Neural networks and their applications. Review of scientific instruments, 65(6), pp.1803-1832.

[4] Coşkun, M., Uçar, A., Yildirim, Ö. and Demir, Y., 2017, November. Face recognition based on convolutional neural network. In 2017 International Conference on Modern Electrical and Energy Systems (MEES) (pp. 376-379). IEEE.

[5] Kasar, M.M., Bhattacharyya, D. and Kim, T.H., 2016. Face recognition using neural network: a review. International Journal of Security and Its Applications, 10(3), pp.81-100.

[6] Khan, S., Javed, M.H., Ahmed, E., Shah, S.A. and Ali, S.U., 2019, March. Facial recognition using convolutional neural networks and implementation on smart glasses. In 2019 International Conference on Information Science and Communication Technology (ICISCT) (pp. 1-6). IEEE.

[7] Nasse, F., Thurau, C. and Fink, G.A., 2009. Face detection using gpu-based convolutional neural networks. In Computer Analysis of Images and Patterns: 13th International Conference, CAIP 2009, Münster, Germany, September 2-4, 2009. Proceedings 13 (pp. 83-90). Springer Berlin Heidelberg.

[8] Novikov, A., Podoprikhin, D., Osokin, A. and Vetrov, D.P., 2015. Tensorizing neural networks. Advances in neural information processing systems, 28.

[9] Wu, J., 2017. Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 5(23), p.495.