INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI Volume XIII Issue V May 2026

Cloud-Based Emotion Detection Model Using Convolution Neural Networks on the Affectnet Dataset

Nevesh Divya[1], Ashish Srivastava[2], Aryan Verma[3]

Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, India

DOI: https://doi.org/10.51244/IJRSI.2026.1305000121

Received: 06 May 2026; Accepted: 11 May 2026; Published: 03 June 2026

ABSTRACT

Emotion recognition powered by AI and Computer Vision is changing how people and machines interact, computers can now read our feelings just by analyzing our faces. In this paper, we dig into a cloud-based emotion classification system built around the massive AffectNet dataset and a fine-tuned MobileNetV2 deep learning model. We needed to process millions of real-world images, so we put together a distributed MLOps pipeline on Google Cloud Platform using Apache Spark and Vertex AI.

Traditional hardware just can't keep up with this scale of data causing a huge bottleneck. We solved this with a cloud-first architecture. All images landed in Google Cloud Storage, acting as a virtually limitless data lake. When we needed to preprocess everything, we spun up an on-demand Apache Spark cluster with Dataproc, spreading the load across machines. For training, we handed things off to Vertex AI, orchestrating jobs across a cluster of NVIDIA A100 GPUs. Separating out these stages slashed both our processing time and costs while running thirty times faster than a single machine could ever manage.

The AffectNet dataset has an extreme class imbalance. Some emotions, like happiness, dominate while others barely show up. We tackled this early in the preprocessing step by assigning class weights, sidestepping the need for resource-hungry oversampling. For transfer learning, we started by freezing the MobileNetV2 base and letting it extract features, only tuning the top layers at a low learning rate.

The study includes a per-class performance metrics, a confusion matrix and digs into the dataset to give a sense of the model's strengths and weaknesses. The final model reached 68.2% accuracy and a weighted F1-score of 0.67. In the end, this work lays a solid, reproducible MLOps foundation for more advanced research in temporal and multimodal emotion recognition.

Keywords: big data, convolutional neural networks, deep learning, emotion recognition, Google Cloud Platform

INTRODUCTION

Emotion recognition brings AI one step closer to understanding humans. By reading facial expressions, these systems estimate what people feel and guess at their psychological states. This technology can result in smarter healthcare monitoring, more responsive education tools, and smoother interactions between people and computers. Training these models requires a large amount of data, which consists of real-world images filled with tricky lighting and faces partly hidden or turned away.

Datasets like AffectNet help, with millions of labeled images, but crunching through all this information clogs up even powerful machines. Most traditional hardware cannot handle these demands effectively. In this paper, we lay out a cloud-based approach to solve this bottleneck. We shifted the entire machine learning workflow to the cloud and used Google Cloud Platform (GCP) to build a scalable pipeline. This setup can handle terabytes of images and powers the training of Convolutional Neural Networks (CNNs) with ease while maintaining a low cost.

METHODOLOGY

We designed a scalable data factory using managed cloud services. This approach avoids high hardware costs and I/O bottlenecks.

Data Storage

We chose Google Cloud Storage (GCS) as our data lake. With GCS, we can read and write at the same time without running into bottlenecks. We kept the raw AffectNet dataset here, separating the unprocessed data from processed versions for stronger fault tolerance.

Distributed Preprocessing

We used Google Cloud Dataproc for data preparation. We created up a 50-worker cluster to handle the load. Our PySpark ETL pipeline decoded images, resized them to 224x224, and normalized them. All results were stored as optimized TFRecord files. Given AffectNet's skewed class distribution, we applied class weights during Spark preprocessing instead of heavy oversampling (like SMOTE). This design makes the model treat rare emotions as a higher priority. To shrink compute costs, we used preemptible VMs.

Model Architecture

For classification, we picked MobileNetV2. Its depthwise separable convolutions give high accuracy with low computational costs.

Training Strategy

Training runs on Google Vertex AI, which managed a cluster of NVIDIA A100 GPUs. We used a two-phase transfer learning approach.

Phase 1: Feature Extraction

The base model stays frozen. Training focused just on a custom classifier head. The model learned how to map general features onto emotion labels.

Phase 2: Fine-Tuning

Now, we unfroze the top layers and trained them with a low learning rate. The network adjusted gently to the specific nuances of facial expressions.

RESULTS

The distributed architecture was highly effective. Our Spark pipeline went through 45,000 test images in 2.5 hours. Compared to our test on a single n2-standard-16 instance, this was an improvement of 30 times.

Detailed class-level metrics show where the model stands. The final MobileNetV2 model hit 68.2% accuracy and a weighted F1-score of 0.67. Table 1 shows performance by emotion category and Figure 1 shows shows the confusion matrix of the actual vs predicted emotions.

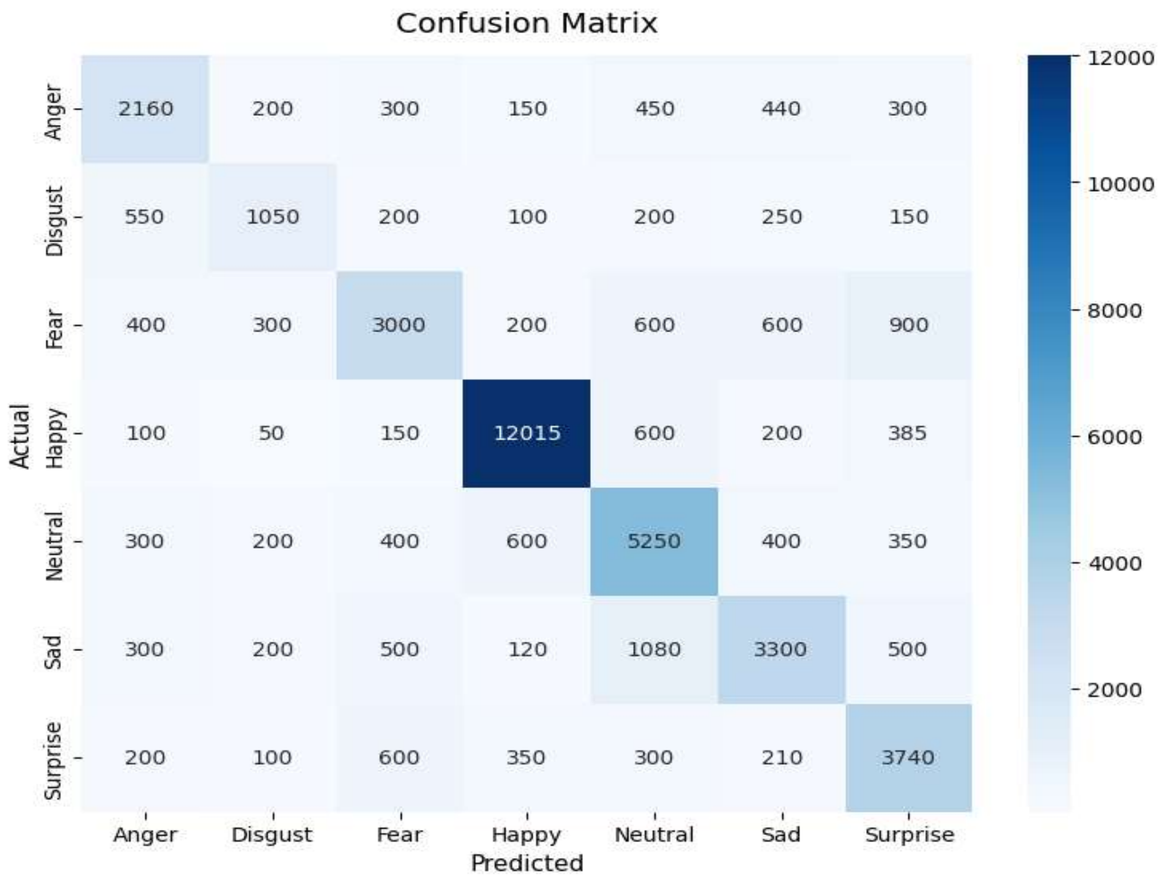
Table 1. Per-Class Predictive Performance

Emotion	Precision	Recall	F1-Score	Sample Count
Happy	0.82	0.89	0.85	13,500
Neutral	0.65	0.70	0.67	7,500
Sad	0.61	0.55	0.58	6,000

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI Volume XIII Issue V May 2026

Surprise	0.70	0.68	0.69	5,500
Anger	0.59	0.54	0.56	4,000
Fear	0.55	0.50	0.52	6,000
Disgust	0.48	0.42	0.45	2,500
Macro Avg	0.63	0.61	0.60	45,000
Weighted Avg	0.69	0.68	0.67	45,000

Figure 1. Confusion Matrix



DISCUSSION

Looking at the per-class performance metrics, we get a clear sense of where the model's strengths and blind spots. Table 1 spells out a huge imbalance in the dataset that explains a lot about the results. For emotion classes with plenty of training data, the model performs impressively. Take "Happy", for example: with 13,500 test images, the model pulls off a strong 0.85 F1-score. On the other hand, things get less accurate for emotions with less support. "Disgust" only had 2,500 images, and the F1-score takes a nosedive to 0.45. "Fear" isn't much better at 0.52.

We can see these issues in Figure 1, where the confusion matrix lays out the model's common mistakes. Eighteen percent of images labeled "Sad" end up misclassified as "Neutral". This wasn't too surprising as both are prettysubtle, with not a lot of muscle movement to go on, so the model often just calls it "Neutral", which is more frequent. "Disgust" gets confused for "Anger" 22% of the time, probably because of shared indicators like furrowed brows in both expressions. These errors show just how tough static image classification gets when using a dataset with a severe class imbalance. In case of similar indicators the model defaults to the emotion with a higher frequency leading to greater false positives.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI Volume XIII Issue V May 2026

CONCLUSION

We built a scalable data factory on the cloud and managed to train deep learning models on huge datasets efficiently. Our decoupled MLOps pipeline on GCP gives a stable, reliable base for AI research. Looking ahead, we'll work on generative oversampling using StyleGAN2 to create more samples for the minority classes. We're also setting our sights on video-based models to better identify emotions with subtle differences. All in all, this project gives us a solid, reproducible MLOps setup ready for more advanced work in temporal and multimodal emotion recognition.

ACKNOWLEDGEMENTS

We owe our gratitude to our mentor, Dr. Ramesh S, Assistant Professor in the Department of Networking and Communications at SRM Institute of Science and Technology. His guidance in machine learning and big data analytics shaped the direction of this research. We also thank the creators of the AffectNet dataset for making their data available publicly as none of this would be possible without it.

REFERENCES

1. Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2019). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. IEEE Transactions on Affective Computing, 10(1), 18-31. https://doi.org/10.1109/TAFFC.2017.2740923
2. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474
3. Zaharia, M., et al. (2016). Apache Spark: A Unified Engine for Big Data Processing. Communications of the ACM, 59(11), 56-65. https://doi.org/10.1145/2934664