International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

A Compact Convolutional Neural Network for Seven-Class Facial Expression Recognition: Per-Class Evaluation and Error Analysis on a Balanced FER2013 Dataset

Ashraf Osman Sayed[1], Khalid Hamid Bilal[2], Yasir Mohamed[3]

[1] *Karary University, Sudan*
[2] *Omdurman Islamic University, Sudan*
[3] *A'Sharqiyah University, Oman*

Abstract - *Compact models are attractive for facial expression recognition, because many applications run on ordinary computers and small devices. This paper reports a small convolutional neural network (CNN) with 80,391 trainable parameters, built from scratch in MATLAB, and describes in detail what it can and cannot do. The network has three convolution blocks (8, 16 and 32 filters) and one hidden layer of 64 units. We trained it for 10 epochs on a class-balanced version of the FER2013 dataset (45,290 images, seven classes) using an 80/20 split, and we evaluated it on the 9,058 validation images. Overall accuracy was 56.47% (95% confidence interval 55.5% to 57.5%), macro F1-score was 55.69% and Cohen's kappa was 0.492. Disgust, Surprise and Happy were recognized best (F1-scores of 87.25%, 68.74% and 68.43%). Sad, Fear, Neutral and Angry were the hardest (F1-scores of 35.67% to 46.68%). These four classes account for 79.5% of all errors, and about two thirds of their errors stay inside the same group. The largest single confusion is Fear predicted as Surprise, which affects 21.7% of Fear images. The model needs about 1.9 million multiply–accumulate operations per image, and it classified an image in 1.59 ms on our machine. It has roughly 250 to 550 times fewer parameters than recent attention and transformer models, but it is also much less accurate. We state the limits of the study plainly: one training run, no fixed random seed, no independent test set, no data augmentation, and a dataset whose balancing was not documented, so duplicate images across the split cannot be ruled out. The results are best read as a transparent baseline for compact FER models.*

Keywords: deep learning; MATLAB; confusion matrix; model complexity; inter-class confusion.

1. INTRODUCTION

Computers that can read facial expressions are useful in teaching, driver monitoring, customer service and care settings [1]. Many of these uses need a model that runs on an ordinary laptop or a small device, so size and speed matter as much as accuracy.

Recent research has pushed accuracy high, but usually with large models. On the RAF-DB benchmark, attention and transformer networks now report close to 90% accuracy or more, and they use tens of millions of parameters [2]–[4]. Some work aims at smaller models [5], [6], but even a recent model built for efficiency, FERMam, still has about 27 million parameters [3]. Much less is written about how a very small network behaves: where it works, where it fails, and what the failures look like.

The FER2013 dataset [7] is a good place to ask this question. It is small in image size (48 × 48 pixels), freely available, and hard. Its labels are noisy, and human agreement with them has been reported at only about 65% to 68% [7]. A later project relabelled the images with ten annotators each, because the original labels were so uncertain [8].

This paper reports one compact CNN and evaluates it in detail. We do not claim a new architecture or a new state of the art. Our aims are more modest:

- to describe a three-block CNN with 80,391 parameters fully enough that others can rebuild it;
- to report its results class by class, with a confusion analysis and confidence intervals;
- to say clearly what the study does not show.

Section 2 describes the data, network and evaluation. Section 3 gives the results and discusses them. Section 4 lists the limitations, and Section 5 concludes.

2. MATERIALS AND METHODS

2.1 Dataset and split

We used a class-balanced version of FER2013 [7] downloaded from Kaggle, where it is publicly available. The images are 48 × 48 pixel grayscale faces in seven classes: angry, disgust, fear, happy, neutral, sad and surprise. The folder holds 6,470 images per class, 45,290 in total. This differs from the original FER2013 release, which has 35,887 images and a very uneven class distribution. We did not balance the data ourselves, and we have no record of how the balanced version was built. In particular, we do not know whether it contains copied or synthetic images. Section 4 returns to this point.

The images were read with MATLAB's imageDatastore, which takes the class label from the folder name. We split each class at random into 80% for training and 20% for validation, using splitEachLabel. This gave 5,176 training and 1,294 validation images per class (36,232 and 9,058 in total). The split was drawn without a fixed random seed, so the exact split cannot be reproduced, although the class proportions can. There is no separate test set. The validation images were used to monitor training and to compute the final results.

2.2 Pre-processing

All images were converted to three-channel RGB by repeating the gray channel and kept at 48 × 48 × 3. MATLAB's augmentedImageDatastore applied this conversion to each mini-batch just before it entered the network, in the same way for training and validation. We used the datastore only for this conversion and did not apply any rotation, shift, flip or other augmentation. The input layer used MATLAB's default zero-centering, which subtracts the mean training image.

2.3 Network architecture

Figure 1 shows the network. It has three convolution blocks. Each block has a 3 × 3 convolution with same padding, batch normalization [9], a ReLU activation [10] and 2 × 2 max pooling with stride 2 [11]. The blocks have 8, 16 and 32 filters, so the feature map shrinks from 48 × 48 × 3 to 6 × 6 × 32. This map is flattened to 1,152 values and passed to a fully connected layer of 64 units with ReLU, then to a fully connected layer of seven units with softmax. The design follows the usual convolutional pattern [12].

The network has 80,391 trainable parameters. The first fully connected layer holds 73,792 of them, which is about 92% of the total. The three convolution blocks together hold only 6,144 parameters. Counting the multiply–accumulate operations (MACs) of the convolution and fully connected layers gives about 1.9 million per image.

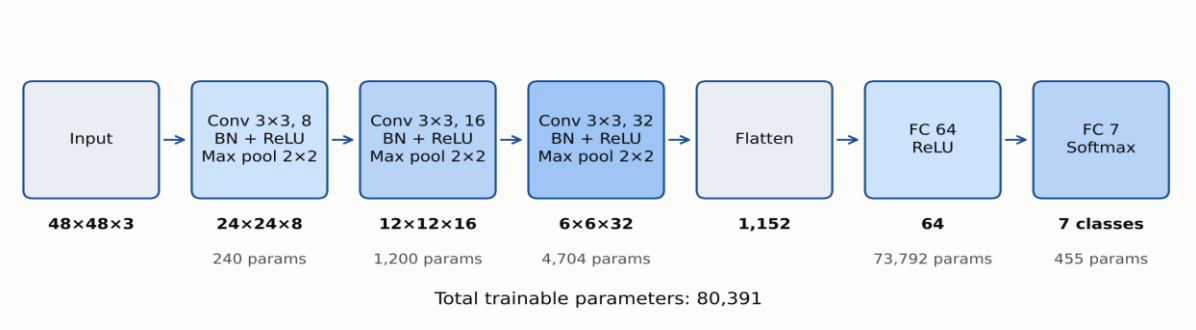


Figure 1. Architecture of the compact CNN, with output shapes and parameter counts per stage.

2.4 Training

The network was trained from scratch with the categorical cross-entropy loss and the Adam optimizer [13]. Table 1 lists the settings. The validation set was evaluated every 10 iterations to draw the training curves. It was not used for early stopping or to pick a checkpoint, and the network from the last iteration was the one evaluated. Training was done in MATLAB with the Deep Learning Toolbox. The MATLAB version, operating system, processor, memory and graphics card were not recorded at the time, so we cannot report them.

Table 1. Training settings.

Setting	Value
Loss	Categorical cross-entropy
Optimizer	Adam
Initial learning rate	0.001
Mini-batch size	32
Maximum epochs	10
Shuffling	Every epoch
Input normalization	Zero-center (MATLAB default)
Data augmentation	None

Setting	Value
Validation during training	Every 10 iterations, for monitoring only
Network used for evaluation	Last iteration

2.5 Evaluation

We computed results from the confusion matrix of the validation set. For each class we treated that class as positive and all others as negative, and counted true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Then:

$$\text{Precision} = TP / (TP + FP) \quad (1)$$

$$\text{Recall} = TP / (TP + FN) \quad (2)$$

$$\text{Specificity} = TN / (TN + FP) \quad (3)$$

$$F1 = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (4)$$

Overall accuracy is the number of correct predictions divided by the number of images. The overall precision, recall and F1-score are macro averages, which means the simple mean over the seven classes. We also report Cohen's kappa [14],

$$\kappa = (p_o - p_e) / (1 - p_e) \quad (5)$$

where p_o is the observed agreement (equal to accuracy) and p_e is the agreement expected by chance. We read kappa with the bands of Landis and Koch [15]. Confidence intervals are 95% Wilson intervals. They treat the validation images as independent, so they describe only the sampling error of this validation set.

3. RESULTS AND DISCUSSION

3.1 Overall results

Table 2 gives the overall results. The network classified 56.47% of the 9,058 validation images correctly. The 95% confidence interval for this accuracy is 55.5% to 57.5%. Every class has the same number of images, so macro recall is exactly equal to accuracy. Chance accuracy for seven balanced classes is 1/7, or 14.3%. The model is far above chance, and its kappa of 0.492 is moderate agreement on the Landis and Koch scale [15]. Still, it gives a wrong label to more than four images in every ten.

Specificity is 92.74%, which looks high. It is easy to reach in a seven-class problem, because each class has six times more negative images than positive ones. We therefore do not take it as evidence of good performance. Accuracy, F1-score and kappa are the fairer measures here.

Table 2. Overall results on the validation set (9,058 images).

Metric	Value
Accuracy	56.47% (95% CI 55.5–57.5%)
Precision (macro)	55.55%
Recall (macro)	56.47%
Specificity (macro)	92.74%
F1-score (macro)	55.69%
Cohen's kappa	0.492
Training time	35,019.79 s (9.73 h)
Prediction time, all 9,058 images	14.38 s (1.59 ms per image)

Training and prediction times were measured on one machine whose specification was not recorded.

3.2 Results by class

Table 3 and Figure 2 show the results for each class. Disgust is recognized best, with an F1-score of 87.25%. Surprise (68.74%) and Happy (68.43%) come next. Angry (46.68%) and Neutral (46.25%) are in the middle. Fear (36.82%) and Sad (35.67%) are the weakest. The 95% intervals for recall are about 2 to 3 points wide on each side, so the order of Disgust, then Surprise and Happy, then the rest is clear. The order inside the groups is not: Surprise and Happy overlap, and so do Fear and Sad.

The Disgust result needs care. In the original FER2013 release Disgust is by far the rarest class, with only a few hundred images. In the balanced version it has 6,470 images like every other class. If the extra images are

copies or small changes of the originals, some of them will be in both the training and validation sets, and the network can score well on them by recognizing images it has already seen. Our data cannot tell us whether this happened. We therefore treat the Disgust score as the one most likely to be too optimistic.

Precision and recall differ for two classes in a way that shows how the network behaves. Surprise was predicted 1,630 times, although there are only 1,294 true Surprise images, so its precision (61.66%) is lower than its recall (77.67%). Neutral was predicted only 985 times, so its recall (40.73%) is lower than its precision (53.50%). The network tends to call uncertain faces Surprise and to avoid calling them Neutral.

Table 3. Results by class on the validation set (1,294 images per class).

Class	Precision (%)	Recall (%)	95% CI for recall (%)	Specificity (%)	F1-score (%)
Angry	45.31	48.15	45.4–50.9	90.31	46.68
Disgust	86.48	88.02	86.1–89.7	97.71	87.25
Fear	38.28	35.47	32.9–38.1	90.47	36.82
Happy	65.31	71.87	69.4–74.3	93.64	68.43
Neutral	53.50	40.73	38.1–43.4	94.10	46.25
Sad	38.30	33.38	30.9–36.0	91.04	35.67
Surprise	61.66	77.67	75.3–79.9	91.95	68.74

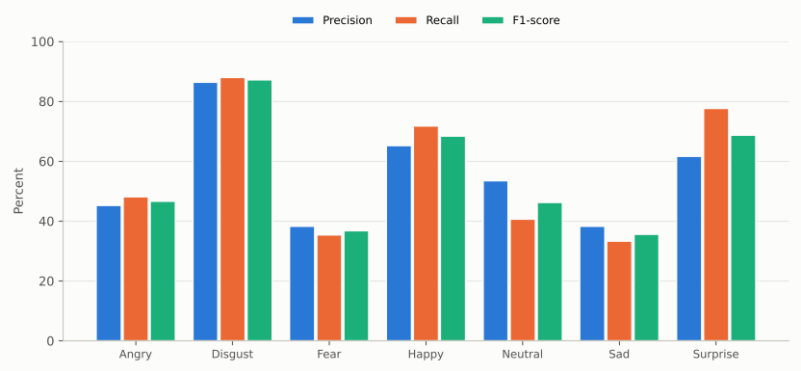


Figure 2. Precision, recall and F1-score for each class.

3.3 Confusion between classes

Figure 3 shows the confusion matrix with each row scaled to 100%, and Table 4 lists the largest off-diagonal entries. The model made 3,943 errors in all. Angry, Fear, Neutral and Sad together are the true class in 3,135 of them (79.5%), and in 1,987 of these 3,135 errors (63.4%) the wrong label is another one of the same four classes. Disgust accounts for only 3.9% of all errors.

The largest single confusion is Fear predicted as Surprise, which affects 21.7% of Fear images. Sad is mistaken for Fear almost as often (19.6%). Neutral and Angry are both often labelled Sad (15.8% and 15.3%). Fear is also mistaken for Angry and Sad (14.6% and 13.3%), and Sad for Angry (14.5%) and Neutral (12.8%). Sad and Fear show the clearest two-way confusion: 253 Sad images were called Fear and 172 Fear images were called Sad.

These are the pairs one would expect to be hard, since several of these expressions involve similar movements of the brows, eyes and mouth. But a confusion matrix cannot show which image features cause an error, and we did not run any feature analysis. We therefore describe the pattern and do not claim to explain it.

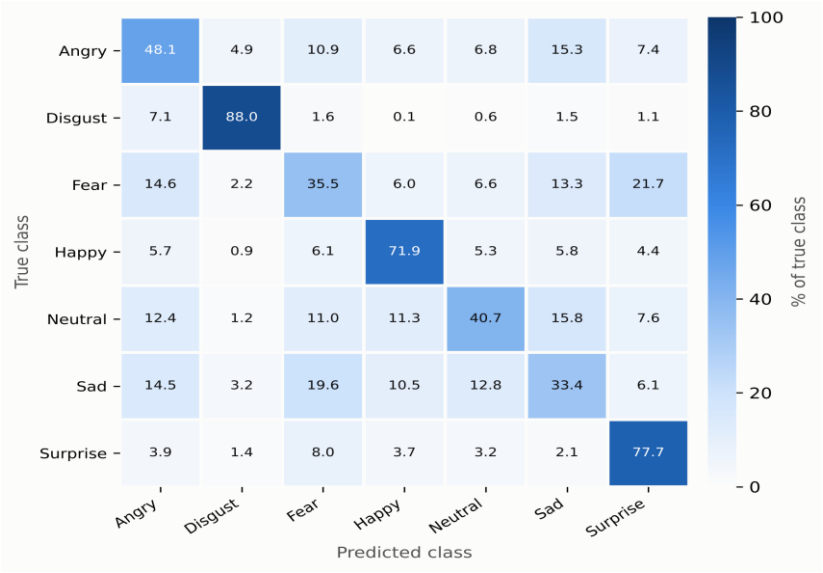


Figure 3. Row-normalized confusion matrix on the validation set. Each row sums to 100%, and the diagonal equals the recall of each class.

Table 4. The six largest confusions (true class predicted as another class).

True class	Predicted as	Images	Share of true class (%)
Fear	Surprise	281	21.7
Sad	Fear	253	19.6
Neutral	Sad	205	15.8
Angry	Sad	198	15.3
Fear	Angry	189	14.6
Sad	Angry	187	14.5

3.4 Model size and speed

The network has 80,391 parameters and needs about 1.9 million MACs per image. For scale, AlexNet, an early large CNN, has about 60 million parameters [16]. Almost all of our parameters are in the first fully connected layer, so a different way of reducing the 6 × 6 × 32 feature map before that layer, such as global average pooling, might shrink the model much further. We have not tested this.

Training took 9.73 hours, which is long for such a small network. A large part of it probably comes from evaluating the 9,058 validation images every 10 iterations during training, but we did not measure this. Classifying all 9,058 validation images took 14.38 seconds, or 1.59 ms per image (about 630 images per second). This was measured with MATLAB's classify function on the whole validation datastore, so it includes loading and resizing the images. Because we did not record the hardware, these times are a rough guide and cannot be compared with other papers.

3.5 Comparison with published results

Published results cannot be compared with ours directly. Our data are a balanced version of FER2013 of unknown construction, we report validation accuracy from one run, and other papers use different datasets, splits and protocols. Table 5 gives published numbers only to show scale.

On FER2013 itself, Jayaraman and Mahendran report 82.89% with a CNN–BiLSTM model that uses dual attention [17], and their model size is not given in the abstract. Anas et al. report 50.77% accuracy and a 29.16% F1-score for an FER2013-trained CNN on the validation set of the ABAW 2020 challenge [18]. Our 56.47% lies between these two, but because the data and protocols differ we draw no ranking from it. Recent attention and transformer models report 89.7% to 92.2% on RAF-DB and 65.7% to 67.5% on seven-class AffectNet, with 19.7 to 43.7 million parameters [2]–[4]. Our network is about 250 to 550 times smaller than these, and, depending on how

each paper counts operations, it needs hundreds to thousands of times fewer operations. It is also far less accurate. Small size has a real cost in accuracy, and this study does not show that the cost could be avoided at this size.

Table 5. Size and reported accuracy of this network and of recent models (accuracy values are on different datasets and are not comparable).

Model	Dataset and accuracy	Parameters (M)	Operations	Source
This work	Balanced FER2013, validation: 56.47%	0.080	1.9 M MACs	This paper
DAN	RAF-DB 89.70%; AffectNet-7 65.69%	19.72	2.23 GFLOPs	[2]
FERMam	RAF-DB 92.13%; AffectNet-7 66.38%	26.99	5.97 GFLOPs	[3]
POSTER++	RAF-DB 92.21%; AffectNet-7 67.49%	43.70	8.40 GFLOPs	[4]
CNN–BiLSTM, dual attention	FER2013: 82.89%	Not given	Not given	[17]
Deep CNN, ABAW 2020	ABAW validation: 50.77%	Not given	Not given	[18]

Values are as stated by the original authors. The paper's own operations are counted as MACs; the other papers state FLOPs, and their counting convention may differ.

4. LIMITATIONS

This study has several limits that readers should keep in mind. First, the dataset was balanced by someone else, and we do not know how. If it contains copied or synthetic images, near-duplicates may lie on both sides of the split, which would make the results look better than they are. The Disgust result is the most exposed to this. Second, there is no independent test set. The validation images served for the final evaluation too, although they were not used to stop training or to choose a model. Third, we trained once, without a fixed random seed, so we cannot say how much the results change between runs. The confidence intervals cover only the sampling error of the validation set, and they assume independent images, which duplicates would violate.

Fourth, we used no data augmentation, which probably lowers the accuracy this network could reach. Fifth, we trained no other model on the same split. We therefore cannot say how much the design choices matter, or how a simpler or a larger network would do on these data. Sixth, we used one dataset and did not test on another. Seventh, we did not record the hardware, so the timings are only rough.

5. CONCLUSION

A three-block CNN with 80,391 parameters reached 56.47% accuracy and a kappa of 0.492 on a balanced seven-class version of FER2013. It recognized Disgust, Surprise and Happy much better than Fear, Sad, Neutral and Angry, and most of its errors were confusions among the last four classes. It is very small and fast, but it is much less accurate than recent larger models. Because of the limits listed in Section 4, the results should be read as a transparent baseline and not as a final measure of what a compact model can do.

The next steps are clear. The original images should be split again with a fixed seed and with a check for duplicates, the training should be repeated over several seeds, and an independent test set should be kept apart. Augmentation, a few same-split baselines and a light attention block can then be tested one at a time to see which of them narrows the gap for Fear, Sad and Neutral.

REFERENCES

[1] S. Li and W. Deng, "Deep facial expression recognition: A survey," IEEE Trans. Affective Computing, vol. 13, no. 3, pp. 1195–1215, 2022, doi: 10.1109/TAFFC.2020.2981446.

[2] Z. Wen et al., "Distract your attention: Multi-head cross attention network for facial expression recognition," Biomimetics, vol. 8, no. 2, art. 199, 2023, doi: 10.3390/biomimetics8020199.

[3] C. Gao et al., "FERMam: A lightweight dual-source and multi-scale fusion framework for facial expression recognition," Scientific Reports, vol. 16, art. 13826, 2026, doi: 10.1038/s41598-026-44396-6.

[4] J. Mao et al., "POSTER++: A simpler and stronger facial expression recognition network," arXiv:2301.12149, 2023.

[5] N. Zhou et al., "A lightweight convolutional neural network for real-time facial expression detection," IEEE Access, vol. 9, pp. 5573–5584, 2021, doi: 10.1109/ACCESS.2020.3046715.

[6] R. Sun et al., "FCCA: Fast Center Consistency Attention for facial expression recognition," Electronics, vol. 14, no. 6, art. 1057, 2025, doi: 10.3390/electronics14061057.

[7] I. Goodfellow et al., "Challenges in representation learning: A report on three machine learning contests," Neural Networks, vol. 64, pp. 59–63, 2015, doi: 10.1016/j.neunet.2014.09.005.

[8] E. Barsoum et al., "Training deep networks for facial expression recognition with crowd-sourced label distribution," in Proc. ACM Int. Conf. Multimodal Interaction (ICMI), 2016, pp. 279–283, doi: 10.1145/2993148.2993165.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. Int. Conf. Machine Learning (ICML), 2015, pp. 448–456.

[10] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in Proc. Int. Conf. Machine Learning (ICML), 2010, pp. 807–814.

[11] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in Proc. Int. Conf. Artificial Neural Networks (ICANN), LNCS vol. 6354, Springer, 2010, pp. 92–101, doi: 10.1007/978-3-642-15825-4_10.

[12] Y. LeCun et al., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations (ICLR), 2015, arXiv:1412.6980.

[14] J. Cohen, "A coefficient of agreement for nominal scales," Educational and Psychological Measurement, vol. 20, no. 1, pp. 37–46, 1960, doi: 10.1177/001316446002000104.

[15] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," Biometrics, vol. 33, no. 1, pp. 159–174, 1977, doi: 10.2307/2529310.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105.

[17] S. Jayaraman and A. Mahendran, "An interactive information based DCNN-BiLSTM model with dual attention mechanism for facial expression recognition," Scientific Reports, vol. 15, art. 26287, 2025, doi: 10.1038/s41598-025-09709-1.

[18] H. Anas et al., "Deep convolutional neural network based facial expression recognition in the wild," arXiv:2010.01301, 2020.