ERROR CORRECTION IN MULTICLASS IMAGE CLASSIFICATION OF FACIAL EMOTION ON UNBALANCED SAMPLES

Andrey A. Lebedev
Lobachevsky State University of Nizhny Novgorod
stasenko@neuro.nnov.ru

Victor B. Kazantsev
Lobachevsky State University of Nizhny Novgorod
Moscow Center for Advanced Studies
kazantsev@neuro.nnov.ru

Sergey V. Stasenko
Lobachevsky State University of Nizhny Novgorod
Moscow Center for Advanced Studies
stasenko@neuro.nnov.ru

October 7, 2025

ABSTRACT

This paper considers the problem of error correction in multi-class classification of face images on unbalanced samples. The study is based on the analysis of a data frame containing images labeled by seven different emotional states of people of different ages. Particular attention is paid to the problem of class imbalance, in which some emotions significantly prevail over others. To solve the classification problem, a neural network model based on LSTM with an attention mechanism focusing on key areas of the face that are informative for emotion recognition is used. As part of the experiments, the model is trained on all possible configurations of subsets of six classes with subsequent error correction for the seventh class, excluded at the training stage. The results show that correction is possible for all classes, although the degree of success varies: some classes are better restored, others are worse. In addition, on the test sample, when correcting some classes, an increase in key quality metrics for small classes was recorded, which indicates the promise of the proposed approach in solving applied problems related to the search for rare events, for example, in anti-fraud systems. Thus, the proposed method can be effectively applied in facial expression analysis systems and in tasks requiring stable classification under skewed class distribution.

Keywords multi-class classification · unbalanced samples · LSTM · attention mechanism · emotions in facial images · classification error correction · rare classes · anti-fraud systems · deep learning · computer vision

1 Introduction

Accurate facial emotion recognition has become a cornerstone in the broader domain of affective computing, which seeks to develop systems capable of recognizing, interpreting, and responding to human emotions using computational methods [1–3]. However, one persistent challenge in facial expression classification is the presence of imbalanced class distributions, where certain emotions are significantly underrepresented [4, 5]. Class imbalance often degrades model generalization, leading to biased predictions toward majority classes and poorer recognition of subtle or rare emotions such as fear or disgust [6].

To address such imbalances, researchers have proposed several strategies. Resampling techniques—such as oversampling minority classes or undersampling majority classes—are commonly employed to rebalance datasets, though they risk introducing redundancy or discarding critical information [7, 8]. Another avenue involves loss reweighting, where algorithms adjust the learning focus by assigning greater weight to underrepresented classes. Methods such as class-balanced loss, focal loss, and label-distribution-aware margins exemplify this approach [9–11]. Hybrid approaches that combine reweighting with resampling have also been shown to improve robustness in imbalanced emotion datasets [12].

Deep neural networks enhanced with attention mechanisms have demonstrated considerable promise in emotion recognition tasks. For instance, models integrating attention within CNN–LSTM architectures allow the system to focus on discriminative facial regions, improving performance [14–16]. Furthermore, attention-based frameworks have been used explicitly to handle uncertainties and class imbalances by emphasizing learning from underrepresented samples [13, 18]. Self-attention and Transformer-based designs have also gained traction for emotion recognition, outperforming traditional CNN-based approaches in scenarios with noisy or imbalanced data [19, 20].

Alongside these approaches, a complementary line of research emphasizes AI error correction without retraining legacy systems. This paradigm, pioneered by Gorban and colleagues, introduces external correctors—lightweight modules that detect risky cases and propose alternative decisions. The theoretical foundation rests on stochastic separation theorems, which demonstrate that in high-dimensional spaces, even simple classifiers (such as Fisher's linear discriminants) can separate error cases from correctly classified ones with high probability [23]. This leads to practical one-shot correction methods capable of improving system reliability without iterative retraining [24]. More recent advances generalize these results to fine-grained, clustered data distributions, showing that families of multi-correctors can handle complex structures and even facilitate new class discovery in high-dimensional settings [25, 26]. These methods have been validated on benchmarks such as CIFAR-10, where external correctors successfully amended errors of deep convolutional networks and supported incremental learning of novel object classes.

Despite these advances, most studies focus on models trained directly on the full emotion set, with balanced or rebalanced inputs, rather than exploring error correction strategies that recover missing or excluded classes. Your novel approach addresses this gap by using an LSTM-based model with attention that is trained on subsets of classes, with subsequent error correction applied to the omitted (seventh) class.

In your experiments, the model is trained on all possible combinations of six out of seven emotion classes, and then evaluated on its ability to correct for the seventh class that was held out during training. Your results show that while the success rate of correction varies per class, even small or rare emotion classes benefit, with key quality metrics—such as precision or recall—improving for those minority classes. These findings complement recent evidence showing that auxiliary learning and reconstruction-based strategies can significantly enhance minority class recognition in affective computing tasks [21, 22].

This finding is particularly significant for real-world applications where detecting rare or atypical emotional signals can be critical—for example, in anti-fraud systems or security contexts. The potential to enhance recognition of these low-frequency classes underscores the practical value of your method in scenarios requiring robust classification under skewed class distributions.

2 Methodology

2.1 The corrector model

Let's consider the problem of multiclass classification in the feature space. Let

$$\mathcal{X} \subseteq \mathbb{R}^n, \qquad \mathcal{Y} = \{1, 2, \ldots, K\}.$$

The basic classifier is defined by the mapping

$$f : \mathcal{X} \to \mathcal{Y}, \qquad x \mapsto f(x),$$

which assigns a class label $f(x) to each point x \in \mathcal{X} \in \mathcal{Y}$.

In this case, the model additionally induces the display

$$\mathbf{e} : \mathcal{X} \to \mathbb{R}^m, \qquad x \mapsto \mathbf{e}(x),$$

where $\mathbf{e}(x)$ is interpreted as an embedding vector obtained during the inference process.

To build a correction mechanism, we introduce the binary mapping

$$g : \mathbb{R}^m \to \{0, 1\}, \qquad \mathbf{z} \mapsto g(\mathbf{z}),$$

where $g(\mathbf{z}) = 0$ corresponds to familiar data, and $g(\mathbf{z}) = 1$ corresponds to unfamiliar (out-of-distribution).

Based on this, we define a generalized classifier with correction:

$$h : \mathcal{X} \to \mathcal{Y} \cup \{\text{new class}\}, \qquad h(x) = \begin{cases} f(x), & \text{if } g(\mathbf{e}(x)) = 0, \\ \text{new class}, & \text{if } g(\mathbf{e}(x)) = 1. \end{cases}$$

Thus, the h function combines two mechanisms:

1. standard classification by means of f on a set of known classes $\mathcal{Y}$;
2. identification of new or missing classes using the g corrector.

This separation makes it possible to formalize the error correction task of a pre-trained system as a combination of the classification task and the task of detecting unknown data. In particular, the proposed approach provides:

- resilience to the appearance of classes that were missing at the learning stage;
- the ability to process unbalanced samples by highlighting "rare" or excluded classes;
- extending the functionality of the original $f$classifier to the more general h system.

2.2 Metrics for evaluating the quality of a proofreader

Let's give the test set $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^{N}$, where $y_j \in \mathcal{Y} = \{1, \ldots, K\}$. Let's denote the predictions of the basic (without correction) model f by $\hat{y}_j^{(f)}$, and the predictions of the system with correction h by $\hat{y}_j^{(h)}$.

For a fixed class $i \in \mathcal{Y}$, we assume

$$J_i = \{j \in \{1, \ldots, N\} : y_j = i\}, \qquad N_i = |J_i|.$$

Let's introduce a set of indexes of class i samples that have been correctly classified by one system or another:

$$A_i = \{j \in J_i : \hat{y}_j^{(f)} = i\} \quad \text{(correct by } f\text{)},$$

$$B_i = \{j \in J_i : \hat{y}_j^{(h)} = i\} \quad \text{(correct by } h\text{)}.$$

We also denote the error sets (for class i):

$$\bar{A}_i = J_i \setminus A_i \quad \text{(errors } f\text{on class } i\text{)}, \qquad \bar{B}_i = J_i \setminus B_i \quad \text{(errors } h\text{on class } i\text{)}.$$

2.2.1 Basic one-class metrics.

First-class accuracy (synonyms of TPR for each class):

$$\mathrm{TPR}_i^{(f)} = \frac{|A_i|}{N_i}, \qquad \mathrm{TPR}_i^{(h)} = \frac{|B_i|}{N_i}.$$

Relative and absolute increment:

$$\Delta_i = \mathrm{TPR}_i^{(h)} - \mathrm{TPR}_i^{(f)}, \qquad R_i = \frac{\mathrm{TPR}_i^{(h)}}{\mathrm{TPR}_i^{(f)} + \varepsilon},$$

where $\varepsilon > 0$ is a small regularization for stability at zero base.

Conservation and harm metrics for already correctly classified samples. The characteristic we are interested in is how the corrector affects those samples that have already been correctly classified by f. We introduce the following concepts.

Retention — the proportion of samples of class i that were correctly recognized by f, which remained correctly recognized even after applying the corrector:

$$\mathrm{Ret}_i \;=\; \frac{|A_i \cap B_i|}{|A_i|} \quad (\text{if } |A_i| > 0).$$

Harm (harm) — the proportion of i class samples correctly recognized by f that became erroneous after correction:

$$\mathrm{Harm}_i \;=\; \frac{|A_i \setminus B_i|}{|A_i|} \;=\; 1 - \mathrm{Ret}_i.$$

Correction gain — the proportion of previously misclassified Class i samples that became correct after correction:

$$\mathrm{Gain}_i \;=\; \frac{|B_i \setminus A_i|}{|\bar{A}_i|} \quad (\text{if } |\bar{A}_i| > 0).$$

2.2.2 Metrics of "spillover" and false positives on other classes.

To take into account how the corrector affects the incorrect reassignment of the label towards the class i, we define

$$\mathrm{Spill}_i \;=\; \frac{|\{j:\; y_j \neq i,\; \hat{y}_j^{(h)} = i\}|}{N - N_i}$$

— the proportion of samples from other classes that were classified as class i by the h system (i.e., "false positive for i"). Accordingly, for f:

$$\mathrm{FPR}_i^{(f)} \;=\; \frac{|\{j:\; y_j \neq i,\; \hat{y}_j^{(f)} = i\}|}{N - N_i}, \qquad \mathrm{FPR}_i^{(h)} \;=\; \frac{|\{j:\; y_j \neq i,\; \hat{y}_j^{(h)} = i\}|}{N - N_i}.$$

False-positive percentage change:

$$\Delta\mathrm{FPR}_i \;=\; \mathrm{FPR}_i^{(h)} - \mathrm{FPR}_i^{(f)}.$$

2.2.3 Summary indicators and standards.

The following aggregates are offered for a summary assessment:

$$\overline{\mathrm{Ret}} = \frac{1}{K}\sum_{i=1}^{K}\mathrm{Ret}_i, \qquad \overline{\mathrm{Harm}} = \frac{1}{K}\sum_{i=1}^{K}\mathrm{Harm}_i,$$

$$\overline{\mathrm{Gain}} = \frac{1}{K}\sum_{i=1}^{K}\mathrm{Gain}_i, \qquad \overline{\Delta\mathrm{FPR}} = \frac{1}{K}\sum_{i=1}^{K}\Delta\mathrm{FPR}_i.$$

You can also weight the frequencies of the classes N_i/N, if desired, to get a "micro" version of the metrics:

$$\mathrm{Ret}^{(\mathrm{w})} = \sum_{i=1}^{K}\frac{N_i}{N}\mathrm{Ret}_i, \quad \text{etc.}$$

2.2.4 Interpretation of metrics.

- Ret_i close to 1 means that the corrector does not spoil the already correctly recognized samples of class i.
- Harm_i demonstrates the share of "collateral damage" for already correct predictions; the lower the better.
- Gain_i shows the corrector's ability to restore previously erroneous examples of this class.
- $\Delta\mathrm{FPR}_i > 0$ indicates an increase in false-positive assignments to class i (side effect).

Statement of the problem

In this paper, we propose a method for error correction in the problem of multi-class classification of facial images with unbalanced samples using an ensemble of a neural network model and a gradient boosting algorithm. The experiments were performed on a dataset containing data on 7 classes of emotional states of people of different ages with a pronounced imbalance in the distribution by classes. The method includes the following stages:

Iterative exclusion of classes. For each experiment, one class is excluded from the training set. The model is trained on the remaining six classes, and the excluded one is used at the inference and correction stage. Thus, a sequential analysis of the model's behavior is carried out in the absence of each of the classes.

Base model: LSTM with an attention mechanism. To process images, a model with an LSTM architecture and an attention mechanism is used, capable of focusing on key areas of the face. Images are pre-encoded by convolutional layers, after which they are passed to the LSTM model.

Feature extraction from latent variables. After the neural network inference, all latent variables are saved — including the outputs of convolutional layers, attention vectors, LSTM hidden states, and other intermediate representations. This allows us to obtain a detailed description of the model's response to each input stimulus.

Training the corrector model (gradient boosting). At the correction stage, the gradient boosting algorithm is used, trained on the full set of extracted features. The goal is to correct the neural network errors on previously unseen examples from the excluded class. Boosting is trained on the neural network inferences and is able to identify patterns that were not taken into account by the first model due to their absence.

Accuracy analysis by classes. After the correction, the accuracy of predictions is estimated for each class separately, including both those presented in training and the previously excluded one. This allows us to determine how effectively each specific class is being restored and to identify the best configurations for correction.

		Data Split			
Label	**Emotion**	**Train**	**Test**	**Correct**	**Sum**
1	Surprise	814	379	426	**1620**
2	Fear	166	88	101	**357**
3	Disgust	449	220	208	**880**
4	Happiness	2974	1506	1477	**5961**
5	Sadness	1216	628	616	**2465**
6	Anger	435	216	216	**873**
7	Neutral	1615	798	791	**3211**
All emotions		**7669**	**3835**	**3835**	**15339**

Table 1: Correspondence between labels and emotions in the RAF-DB dataset (basic emotions)

Data

For analysis, the RAF-DB (Real-world Affective Faces Database) dataset was selected, containing data on the emotions of people of different ages [**?**].

Examples of images are shown in Fig 1. The dataset contains 15,339 facial images labeled with basic and composite emotions by 40 independent annotators. The label mapper, which shows the face label and the emotion it encodes, is shown in 1. The images are highly diverse in features such as age, gender, ethnicity, head rotation, lighting conditions, and the presence of partial occlusions (e.g. glasses, beard, or self-occlusion). In addition, the images may contain filters and special effects, which makes the dataset particularly suitable for emotion recognition tasks in near-real-world settings.

Figure 1: Examples of images labeled with different emotions in the sample, according to the labeling shown in Table 1.

To conduct experiments according to the described methodology, it is necessary to select three random subsets from the initial data:

- **Train** — data for training the LSTM model regardless of the class configuration;
- **Correct** — data for training the corrector model;
- **Test/Val** — data for testing and assessing the accuracy of the models.

Distributions of the proportions of classes in each subset are given in figure 2.

3 Technical implementation of the experiment

To solve the problem of classifying emotions in facial images, a custom neural network model was developed that combines convolutional layers for feature extraction, a recurrent component (LSTM) for modeling the spatial-sequential structure, and an attention mechanism for identifying the most informative features.

The model consists of the following main components:

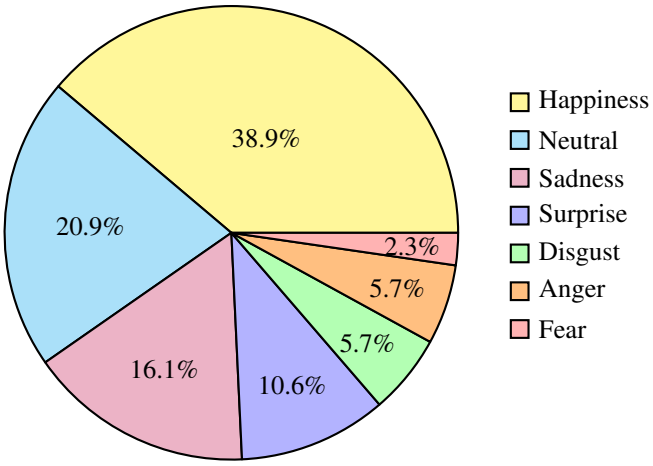

Figure 2: Pie chart showing the proportion of classes in the initial sample.

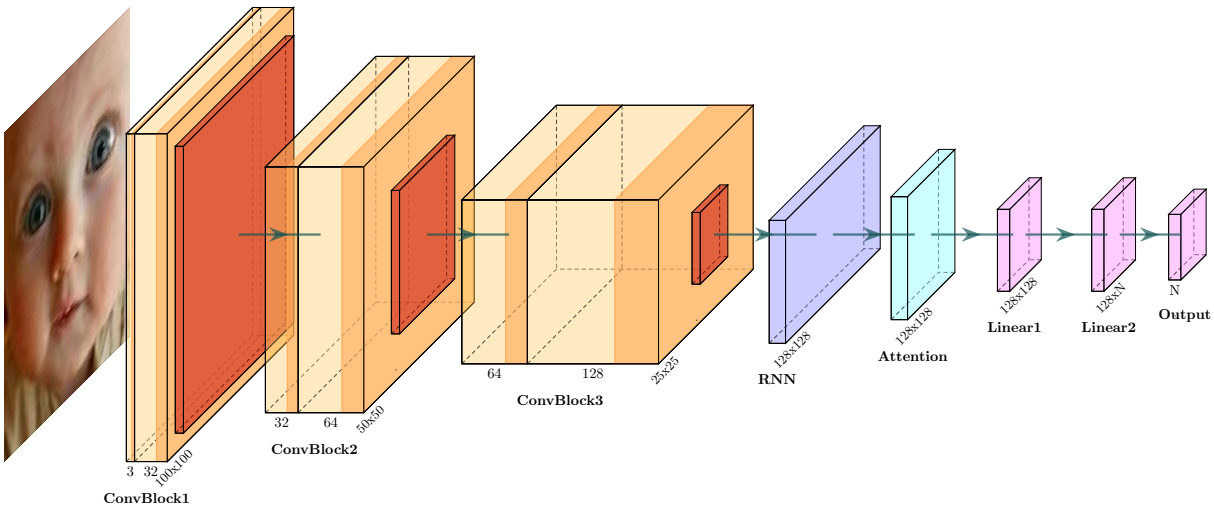

Figure 3: Scheme of the developed convolutional neural network model with memory and attention

3.1 Feature extraction block / Convolutional block (`conv_layers`)

Consists of three convolutional blocks, each of which includes:

- convolutional layer (`Conv2d`) with kernel 3×3, stride 1 and padding 1;
- batch normalization (`BatchNorm2d`);
- ReLU activation function;
- pooling layer `MaxPooling` with kernel 2×2.

Layer parameters:

- Conv1: `in_channels=3`, `out_channels=32`;
- Conv2: `in_channels=32`, `out_channels=64`;
- Conv3: `in_channels=64`, `out_channels=128`.

At the output of the block, spatial feature maps of dimension $[B, 128, H', W']$, where B is the batch size, are formed.

3.2 NN block

3.2.1 Recurrent Block (`rnn`)

The output of the convolutional block is transformed into a sequence: the tensor is transformed into the form $[B, T, D]$, where $T = H' \cdot W'$, and $D = 128$. On this sequence, an LSTM is applied with the following parameters:

- input and hidden state size: 128;
- `batch_first=True`.

This allows the model to capture spatial dependencies between different regions of the image.

3.2.2 Attention Mechanism (`attention`)

The output of the LSTM is a multi-headed attention mechanism (`MultiheadAttention`):

- key, query, and value dimensions: 128;
- one projection layer (`out_proj`) is used at the output.

The attention mechanism allows the model to focus on the most significant areas of the face when making a classification decision.

3.3 Classification head / Fully connected classifier (`fc_layers`)

The attention result is aggregated and fed to the fully connected part of the model:

- `Linear(128 → 128)` with ReLU activation;
- `Dropout(p=0.5)` for regularization;
- `Linear(128 → 7)` is the output layer for classification by N emotions.

3.4 Loss function

Since there is a pronounced imbalance between the classes in the training set, a weighted loss function is used. The class weights are calculated as the ratio of the total number of examples to the number of examples in each class:

$$w_i = \frac{N}{n_i},$$

where N is the total number of samples, n_i is the number of samples of the i-th class.

Next, the cross-entropy loss function is used:

$$\mathcal{L} = -\sum_{i=1}^{C} w_i \cdot y_i \log(\hat{y}_i),$$

where C is the number of classes, y_i is the true label, $\hat{y}_i$ is the model output (softmax probability) for class i, and w_i is the class weight.

3.5 Early stopping mechanism

To determine the criterion for early learning stoppage, we tried several different rules based on the average roc-auc-score, loss function, and average accuracy. It has been empirically found that the best rule for stopping learning in the proposed architecture is:

- loss function on train set → min;
- accuracy on the validation set → max.

3.6 Model training.

The results of a study of a descriptive neural network in conjugation of both grades 6 and 7 showed, in the form of a segmented metric, the accuracy of determining classes in the table 4.

Accuracy is calculated as the proportion of correctly predicted labels:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(y_i = \hat{y}_i),$$

where $\mathbb{I}$ is the indicator function.

3.7 Extraction of hidden features

For correction, a method of repeated inference of the model on additional images that were not included in the training and validation samples was used. During the inference process, the values of all intermediate representations (feature maps and hidden states) were saved at the stages:

- output from convolutional layers;
- output from the LSTM layer;
- after applying the attention mechanism (Multihead Attention);
- output from the fully connected layer before the final classification.

Thus, for each image, a vector of hidden features was formed, containing multi-level information about its spatial and semantic content.

4 Mechanism for Correcting Neural Network Predictions

After completing the training of the basic neural network model, an additional stage of analysis and correction of predictions was implemented in order to improve the quality of classification under uncertainty on new data.

4.1 Training a Gradient Boosted Model

The generated latent feature vectors were used as input for a gradient boosting model (e.g., `XGBoost` or `LightGBM`). The target variable was the true class label of the image.

A special feature of the approach is that the boosting model is trained on examples that were not used to train the original neural network. This allows boosting to identify patterns in the latent representations that were not covered by LSTM training, and thus compensate for possible gaps in the generalization ability of the base model.

4.2 Training the corrector.

To train the corrector, we used the XGBoost implementation with parameters selected empirically.

4.3 Using the Corrector

After training, the corrector model was used as an additional filter: during inference, the main model passes the hidden features to boosting, and, in case of a high error probability ($g(\mathbf{z}_i) < \tau$), the prediction is either corrected or not depending on the decision policy.

4.4 Purpose of the method

Thus, gradient boosting acts as an external evaluator over the outputs of the neural network. It can:

- improve the accuracy of class recognition on previously unseen data;
- identify areas of low confidence in the neural network;
- use more flexible dependencies between latent features and class labels.

This approach can be viewed as a type of stacking, where boosting serves as a meta-model trained on the embeddings of the base neural network.

5 Results

This section presents the quantitative results of applying the proposed classification error correction method. The analysis was carried out using the metrics *Retention*, *Harm*, changes in *Gain*, $\Delta\mathrm{FPR}$, as well as the integral performance characteristics listed below.

Retention							
	Corrector						
Label	1	2	3	4	5	6	7
1	1.000	1.000	1.000	0.963	0.995	0.997	0.971
2	0.898	1.000	0.977	0.989	1.000	0.977	1.000
3	1.000	1.000	1.000	0.977	1.000	1.000	0.945
4	0.999	0.999	0.999	1.000	0.997	0.999	0.975
5	0.992	1.000	0.995	0.876	1.000	1.000	0.881
6	0.981	0.995	1.000	0.903	0.968	1.000	0.977
7	0.982	0.999	0.997	0.955	0.957	1.000	1.000
Average	0.973	0.999	0.996	0.940	0.995	0.996	0.963

Harm							
	Corrector						
Label	1	2	3	4	5	6	7
1	0.000	0.000	0.000	0.037	0.005	0.003	0.029
2	0.102	0.000	0.023	0.011	0.000	0.023	0.000
3	0.000	0.000	0.000	0.023	0.000	0.000	0.055
4	0.001	0.001	0.001	0.000	0.003	0.001	0.025
5	0.008	0.000	0.005	0.124	0.000	0.000	0.119
6	0.019	0.005	0.000	0.097	0.032	0.000	0.023
7	0.018	0.001	0.003	0.045	0.043	0.000	0.000
Average	0.021	0.001	0.005	0.048	0.012	0.004	0.036

Table 2: Values of *Harm* and *Retention* metrics for all classes. The Harm metric reflects the proportion of examples that were correctly classified by the original model, but became erroneous after applying the corrector. The Retention metric shows the proportion of correct predictions that remained after the correction. The values on the main diagonal are underlined and correspond to preserving/distorting predictions within the same class. The Retention and Harm metrics are also highlighted in red, which are higher and lower than the average, respectively.

Label	$\overline{\Delta\mathrm{FPR}}$	$\overline{\mathrm{Gain}}$
1	0.015	0.517
2	0.001	0.239
3	0.029	0.482
4	0.095	0.811
5	0.023	0.242
6	0.002	0.204
7	0.072	0.590

Table 3: Summary metrics of proofreader quality for each class. $\overline{\Delta\mathrm{FPR}}$ — the average change in the false positive error after applying the corrector. $\overline{\mathrm{Gain}}$ — the proportion of previously erroneous examples of the corrected class that were classified correctly.

From the results shown in the 2 table, it can be seen that the values of *Retention* remain high (close to unity) in all cases, which indicates that most of the correct predictions of the basic model are preserved. The values of *Harm*, on the contrary, record cases when the corrector introduces distortions. From the table 2 it can be seen that the corrector does the most harm to the recognition of Fear(2) when correcting Surprise(1), Sadness(5) and Anger(6) when correcting Happiness(4) and Sadness(5) when correcting Neutral(7)

Summary indicators of the corrector's effectiveness are given in the table 3. There is a positive trend for all classes *Gain*, which confirms the correctors ability to improve the completeness of the classification of the excluded class. The growth of $\Delta\mathrm{FPR}$ remains relatively moderate, which makes the increase in *Gain* statistically significant. The

greatest increase in quality is observed for «Happiness»(4), «Surprise»(1), «Disgusting»(3), «Neutral»(7) and even worse «Fear»(2), «Sadness»(5), «Anger»(6).

The table 4 shows the results of experiments to exclude classes from the training sample. In different cases, the correction allows for varying degrees of partial compensation for the loss of information about the excluded class.

Predict	**Correct**							**Not correct**	**P**
Label	**1**	**2**	**3**	**4**	**5**	**6**	**7**		
1	0.51	0.77	0.79	0.84	0.75	0.76	0.80	0.78	0.55
2	0.55	0.24	0.06	0.43	0.47	0.45	0.39	0.43	0.56
3	0.38	0.50	0.48	0.98	0.98	0.97	0.95	0.74	0.65
4	0.99	0.99	0.99	0.77	0.86	0.90	0.92	0.90	0.86
5	0.69	0.71	0.72	0.71	0.24	0.72	0.80	0.78	0.31
6	0.68	0.66	0.63	0.63	0.73	0.20	0.63	0.67	0.30
7	0.77	0.76	0.73	0.79	0.87	0.71	0.59	0.90	0.66

Table 4: Comparison of classification accuracy when using a corrector for different combinations of training classes. Values on the main diagonal (errors within its own class) highlighted in red, the maximum values by row (the best prediction option for this class) are green. The last column shows the accuracy of the original model without correction.

For a formal assessment of the corrector's contribution to the final quality of the classification, we introduce corrector power indicator P, which is defined as the ratio of classification accuracy with correction to accuracy without correction:

$$P = \frac{\text{Accuracy}_{\text{with corr}}}{\text{Accuracy}_{\text{without corr}}}.$$

The table 4 shows the values of the strength indicator of the corrector P. The best result was recorded for the class «Happiness» ($P = 0.86$), while for the classes «Sadness» and «Anger» the values P is significantly lower ($P \approx 0.30$). This indicates a pronounced class dependence of the method's effectiveness: for some emotions, correction successfully restores accuracy, for others, the increase is lower.

Thus, the error correction method is a promising tool. It is capable of increasing the stability of multiclass classifiers in conditions of incomplete and unbalanced samples, but its effectiveness depends on the class structure and requires further research.

6 Discussion

However, in the general case, correction turns out to be less effective than complete retraining and does not allow reaching the level of a fully trained model. In turn, building a corrector is computationally much easier than completely retraining the model, as well as the proposed method allows you to create many correctors that implement a complex, nonlinear stack model. Further development of the approach may go in the direction of:

- for using more compact embeddings;
- scaling experiments on large samples to increase stability;
- for developing adaptive correctors that take into account the semantic and statistical relationships of classes.

7 Conclusion

The results show that the proposed error correction method can significantly improve the quality of recognition of excluded classes, while maintaining high accuracy on already known data. However, a significant dependence of efficiency on the nature of a particular class has been revealed.

High values of *Retention* indicate that the basic structure of the model's predictions is preserved., and the risk of quality deterioration for the initial classes remains low. At the same time, the values *Harm* and ΔFPR demonstrate, that the corrector can introduce undesirable distortions, especially in the case of emotions that are similar in feature space. Thus, the method has a positive effect, but requires careful adjustment.

Experiments on class exclusion show that the corrector works especially successfully for classes with pronounced interclass similarity (for example, «Happiness»), whereas for classes less related to others (for example, «Sadness» and «Anger»), the effectiveness is noticeably lower.

8 Acknowledgements

The work was carried out with the support of the federal assignment of the Ministry of Science and Higher Education of the Russian Federation, project No. FSMG-2024-0047.

References

[1] Tao, J. & Tan, T. Affective computing: A review. International Journal Of Automation And Computing. **2**, 302-312 (2005)

[2] Calvo, R. & D'Mello, S. Affect detection: An interdisciplinary review of models, methods, and their applications. IEEE Transactions On Affective Computing. **1**, 18-37 (2010)

[3] Liu, Y. & Fu, G. Emotion recognition by deeply learned multi-channel textual and EEG features. Future Generation Computer Systems. **119** pp. 1-6 (2021)

[4] Koelstra, S., Muhl, C., Soleymani, M., Lee, J., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A. & Patras, I. DEAP: A database for emotion analysis using physiological signals. IEEE Transactions On Affective Computing. **3**, 18-31 (2012)

[5] Sariyanidi, E., Gunes, H. & Cavallaro, A. Automatic analysis of facial affect: A survey of registration, representation, and recognition. IEEE Transactions On Pattern Analysis And Machine Intelligence. **37**, 1113-1133 (2015)

[6] Goodfellow, I., Erhan, D., Carrier, P., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D. & Others Challenges in representation learning: A report on three machine learning contests. International Conference On Neural Information Processing. pp. 117-124 (2013)

[7] He, H. & Garcia, E. Learning from imbalanced data. IEEE Transactions On Knowledge And Data Engineering. **21**, 1263-1284 (2009)

[8] Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. SMOTE: Synthetic minority over-sampling technique. Journal Of Artificial Intelligence Research. **16** pp. 321-357 (2002)

[9] Cui, Y., Jia, M., Lin, T., Song, Y. & Belongie, S. Class-balanced loss based on effective number of samples. Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition. pp. 9268-9277 (2019)

[10] Lin, T., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. Proceedings Of The IEEE International Conference On Computer Vision. pp. 2980-2988 (2017)

[11] Khan, S., Hayat, M., Zamir, S., Shen, J. & Shao, L. Striking the right balance with uncertainty. IEEE Transactions On Pattern Analysis And Machine Intelligence. **41**, 2995-3007 (2019)

[12] Huang, C., Li, Y., Change Loy, C. & Tang, X. Deep imbalanced learning for face recognition and attribute prediction. IEEE Transactions On Pattern Analysis And Machine Intelligence. **42**, 2781-2794 (2019)

[13] Altalhan, M., Algarni, A. & Alouane, M. Imbalanced data problem in machine learning: A review. IEEE Access. (2025)

[14] Hans, A. & Rao, S. A CNN-LSTM based deep neural networks for facial emotion detection in videos. International Journal Of Advances In Signal And Image Sciences. **7**, 11-20 (2021)

[15] Rajpoot, A., Panicker, M. & Others Subject independent emotion recognition using EEG signals employing attention driven neural networks. Biomedical Signal Processing And Control. **75** pp. 103547 (2022)

[16] Wang, K., Peng, X., Yang, J., Meng, D. & Qiao, Y. Region attention networks for pose and occlusion robust facial expression recognition. IEEE Transactions On Image Processing. **29** pp. 4057-4069 (2020)

[17] Wang, L., Zhang, L., Qi, X. & Yi, Z. Deep attention-based imbalanced image classification. IEEE Transactions On Neural Networks And Learning Systems. **33**, 3320-3330 (2021)

[18] Saurav, S., Saini, R. & Singh, S. An integrated attention-guided deep convolutional neural network for facial expression recognition in the wild. Multimedia Tools And Applications. **84**, 10027-10069 (2025)

[19] Huang, Q., Huang, C., Wang, X. & Jiang, F. Facial expression recognition with grid-wise attention and visual transformer. Information Sciences. **580** pp. 35-54 (2021)

[20] Daihong, J., Lei, D. & Jin, P. Facial expression recognition based on attention mechanism. Scientific Programming. **2021**, 6624251 (2021)

[21] Yi, W., Sun, Y. & He, S. Data augmentation using conditional GANs for facial emotion recognition. 2018 Progress In Electromagnetics Research Symposium (PIERS-Toyama). pp. 710-714 (2018)

[22] Chen, H., Guo, C., Li, Y., Zhang, P. & Jiang, D. Semi-supervised multimodal emotion recognition with class-balanced pseudo-labeling. Proceedings Of The 31st ACM International Conference On Multimedia. pp. 9556-9560 (2023)

[23] Gorban, A., Grechuk, B. & Tyukin, I. Stochastic separation theorems: how geometry may help to correct AI errors. Notices Of American Mathematical Society., 25-33 (2023)

[24] Gorban, A., Golubkov, A., Grechuk, B., Mirkes, E. & Tyukin, I. Correction of AI systems by linear discriminants: Probabilistic foundations. Information Sciences. **466** pp. 303-322 (2018)

[25] Gorban, A., Grechuk, B., Mirkes, E., Stasenko, S. & Tyukin, I. High-dimensional separability for one-and few-shot learning. Entropy. **23**, 1090 (2021)

[26] Grechuk, B., Gorban, A. & Tyukin, I. General stochastic separation theorems with optimal bounds. Neural Networks. **138** pp. 33-56 (2021)