Sentiment Detection Using Neutral Network

Author's Name:

1.Miss. Nikita Balaso Patil
PG Student, Padmbhooshan Vasantraodada Institute of Technology, Budhgaon, Sangli.
2.Dr. Sonali S Sankpal
Assistant Professor, Padmbhooshan Vasantraodada Institute of Technology, Budhgaon, Sangli.

Abstract— Getting machines to understand how humans feel is one of the most interesting problems in computer science today. As people spend more time communicating online through social media, video chats, and messaging apps there is a growing need for technology that can pick up on emotional signals from different types of input. This paper introduces a system that looks at two things at the same time: the words a person writes and the expression on their face. For written text, we built a model called SENN that combines two well-known building blocks a BiLSTM and a CNN so that it can understand both the overall meaning of a sentence and the emotional weight of individual words. For facial images, a separate CNN examines photos and groups them into emotion categories based on visual patterns it has learned. Our tests show that using both types of input together gives better results than using either one alone. Both of our deep learning models also beat older, more traditional methods by a clear margin. We also discuss the real challenges the system still faces such as unclear or sarcastic language, bad lighting in photos, and the fact that some emotions look very similar to each other. The paper ends with ideas for future improvements, including building systems that adjust to individual users and applying this work to areas like mental health support and road safety.

Keywords: Emotion Recognition, Deep Learning, CNN, BiLSTM, SENN, Text Analysis, Facial Expression, Multimodal AI

1. INTRODUCTION

Emotions play a huge role in how we talk to each other and how we make choices every day. If we want computers to truly help people as tutors, health assistants, or customer support agents they need to do more than just understand the words we say. They need to understand how we feel when we say them. This is a hard problem that sits right at the crossroads of two big fields in AI, Natural Language Processing, which deals with understanding text, and Computer Vision, which deals with understanding images. In the early days, emotion detection systems worked by comparing words against a fixed dictionary of emotional terms. These were simple and easy to understand, but they failed quickly when people used sarcasm, slang, or unusual phrasing. Then came statistical approaches like Support Vector Machines and Naive Bayes classifiers, which were better but still required a lot of manual work to prepare features. Deep learning changed all of this. Modern neural networks can figure out patterns on their own, directly from raw text or images, without needing humans to point out which features matter. This has led to dramatic improvements in accuracy across the board. Even so, systems that only look at one type of input have clear blind spots. A text-only model can't see the smirk that reveals a joke. A vision-only model can't catch the bitterness hidden in a carefully written sentence. Combining both makes a lot of sense, and that is exactly what this paper is about.

2. DATA USED IN THIS STUDY

2.1 Where the Data Came From

This study relies on two separate datasets one made up of written text and one made up of photos of faces. Both were chosen because they include real-world variety, not just tidy samples from a controlled lab environment.

For the text side, we gathered posts and messages from social media, personal blogs, conversations, and news headlines. We made a point of including many different writing styles because emotional language changes a lot depending on where it comes from a tweet sounds very different from a news article. Each text sample was tagged with one of six emotions: joy, sadness, anger, fear, surprise, or disgust. Human experts did the labeling following a clear set of rules.

For the face side, we used a well-known collection of grayscale facial images around 35,000 in total each sized at 48 by 48 pixels. This dataset adds a seventh category called neutral, for faces that don't show any strong feeling. It includes people of different ages and genders, which helps the model work well for a

Dataset	Where It Comes From	Size	Emotion Labels	Purpose
Text	Social media, blogs, news	Multiple sources	Joy, Sadness, Anger, Fear, Surprise, Disgust	Text emotion detection
Faces	FER-style image dataset	35,000 images	Happy, Sad, Angry, Fear, Surprise, Disgust, Neutral	Facial emotion recognition

wider range of people.

2.2 Getting the Data Ready

Before any data goes into a neural network, it needs to be cleaned up. Raw input is messy, and that messiness can cause a model to learn the wrong things.

For the text data, we did the following:

- Removed special characters, numbers, and unnecessary punctuation
- Turned all text to lowercase so that 'Happy' and 'happy' are treated the same way
- Split sentences into individual words
- Optionally removed very common words like 'the' or 'is' that don't carry emotional meaning
- Converted words into number-based vectors using pre-trained tools like Word2Vec, GloVe, or Fast Text. Fast Text worked especially well for unusual or made-up words because it breaks words into smaller pieces

For the face images, the steps were:

- Made sure all images were in the correct format and looked consistent
- Scaled pixel values to fall between 0 and 1, which helps the model train more smoothly
- Confirmed every image was exactly 48 by 48 pixels
- Turned emotion names into numbered categories the model can process
- Split all data into a training set and a validation set so results could be tested fairly

3. How the System Was Built

The system has two separate parts one that handles text and one that handles faces. After each part does its job, their results are combined to make the final emotion decision.

3.1 The Overall Flow

Here is how the whole pipeline works from start to finish:

- Collect raw text and face images
- Clean and prepare each type of data separately
- Run each through its own deep learning model to extract useful features
- Train both models on their labeled data
- Merge the two sets of features to produce the final emotion prediction.

The key idea is that text and faces each give you different but related clues about how someone is feeling. When one source is unclear, the other can fill in the gap.

3.2 The Text Model: SENN

SENN is built around the idea that understanding emotional language requires two different kinds of reading: understanding the overall flow of a sentence, and spotting specific emotional trigger words or phrases.

At the start, each word in the text is turned into two types of number vectors one that captures its general meaning in context, and one that captures its emotional tone. This gives the model a richer picture to work with than a single vector could provide.

The BiLSTM component reads the text in both directions at once forward and backward so it can understand how each word relates to everything around it. This is particularly helpful for catching negations, like 'not happy,' where the meaning of a word flips because of another word nearby.

At the same time, a CNN runs its own analysis, using small sliding windows of different sizes to look for patterns combinations of words that typically signal a certain emotion. It picks out the strongest signal it finds across the whole sentence.

The outputs from both the BiLSTM and the CNN are then combined into one big list of numbers, passed through a final layer, and turned into a probability for each of the six emotions. The emotion with the highest probability becomes the prediction.

3.3 The Face Model: CNN

The face recognition part works differently because the input is an image, not a sequence of words. A deep CNN is used to look at faces and identify emotional expressions by building up an understanding of visual features layer by layer.

The first few layers of the network pick up basic

visual details like edges and shapes. Deeper layers combine those into more complex patterns the curve of a mouth, the position of eyebrows, or the tension in the muscles around the eyes. The model learns which combinations of these patterns belong to which emotions.

After each convolutional layer, a function called ReLU helps the network handle more complex patterns, and a pooling step reduces the size of the image so computation stays manageable. This pooling also makes the model less sensitive to small shifts in position, so the same expression is recognized whether the face is slightly left or right of center.

At the end, the visual features are flattened and passed through a few connected layers before reaching a final output layer that assigns probabilities to all seven emotion categories.

3.4 Combining the Two Models

We tried two different ways of merging the text and face models. In the first approach, we joined the feature lists from both models before making any prediction, letting the network learn how they relate to each other. In the second approach, each model made its own prediction, and then those predictions were averaged or weighted to get a final answer.

Both methods improved on using either model alone. The first approach generally worked better when the text and image were recorded at the same time for example, a video clip with matching audio transcript because the information was naturally connected.

3.5 Training Details

Both models were trained using the same general setup. The loss function used was Categorical Cross-Entropy, which is the standard choice when you have multiple categories to predict. The

Adam optimizer was used to adjust the model's weights during training, with the learning rate gradually reduced over time. Each training step processed batches of 64 to 128 samples. Training continued until the model stopped improving on the validation set. Dropout layers were added to prevent the models from memorizing the training data too closely. Adam optimizer was used to adjust the model's weights during training, with the learning rate gradually reduced over time. Each training step processed batches of 64 to 128 samples. Training continued until the model stopped improving on the validation set. Dropout layers were added to prevent the models from memorizing the training data too closely.

3.6 How We Measured Performance

We used four measures to evaluate the models: accuracy, precision, recall, and F1-score. Using all four together gives a much more complete picture than accuracy alone, especially since some emotions appeared far more often than others in the data.

4. Results

4.1 Text Results

When tested across a range of text sources social media, news, and conversations SENN consistently beat all the traditional machine learning methods. Models like Naive Bayes, SVM, and logistic regression scored noticeably lower on both accuracy and F1.

SENN was most confident when classifying clear-cut emotions like joy and anger, where the words people use tend to leave little room for doubt. It had more trouble with fear, which often overlaps linguistically with other negative emotions. Switching from GloVe to FastText as the word embedding tool led to a noticeable improvement, especially for informal text with unusual spellings or slang.

4.2 Face Results

The face recognition model reached validation accuracy somewhere between 65% and 75%, depending on which subset of the data was being tested. It did best on happy and surprised expressions, which tend to involve obvious and distinct facial movements. Fear and sadness were harder to tell apart, since both often involve similar features like downturned mouths and furrowed brows.

The training process went smoothly accuracy improved steadily and the loss kept dropping without any sudden validation accuracy, which suggests a mild tendency to over-rely on training examples. This could be improved with more varied training images.

4.3 Comparison Table

Model	Accuracy	Precision	Recall	F1
SENN	High	High	High	High
CNN	Moderate	Moderate	Moderate	Moderate
Traditional ML	Low–Moderate	Moderate	Moderate	Moderate

Table 2: How the different models compare across performance measures

Model	Technique	Task	What Makes It Different
SENN	BiLSTM + CNN	Text emotion recognition	Reads sentences both ways and scans for emotional word patterns at the same time
Deep CNN	Stacked conv layers	Facial emotion recognition	Learns visual features step by step, from basic edges to full expressions
Traditional ML	SVM, Naive Bayes, LR	Baseline comparison	Easy to understand and fast to train, but limited by hand-made features

Table 3: Overview of all models used in this study

5. Challenges We Ran Into

No system is perfect, and being honest about its weaknesses is just as important as reporting what it does well. Here are the four main problems we encountered:
Language is tricky. Text can mean completely different things depending on tone and context. Sarcasm, irony, and understatement are very hard for any model to get right because they require understanding things that aren't said out loud. Short, incomplete messages like many social media posts make this even harder because so much context is missing.
Real-world photos are messy. The face model was trained on fairly consistent images, but in the real world, faces appear under all kinds of conditions different lighting, at angles, partially hidden by glasses or masks, or blurry from a low-quality camera. The model struggled when things strayed too far from what it had seen during training. Some emotions come up much less often than others. In everyday life, people express joy and surprise far more than disgust or fear. When a model sees a lot more examples of some classes than others, it naturally gets better at the common ones and worse at the rare ones. Special techniques are needed to compensate for this imbalance.
Models trained in one setting don't always transfer well. A model that works well on carefully put-together research data might fail in a real clinical or workplace setting where the data looks quite different. Bridging this gap requires either collecting more diverse training data or building in ways for the model to adapt to new environments.

6. Conclusion

The overall takeaway from this study is clear combining text and visual inputs makes emotion recognition more reliable, and deep learning is a big step forward compared to older techniques.
For text, the combination of sequential reasoning (BiLSTM) and pattern detection (CNN) does a better job than either alone. There is still room to grow, especially for emotions that are hard to detect from a single sentence. Future systems might look at whole conversations rather than isolated messages.
For faces, the CNN handles common expressions well but struggles with ones that look similar to each other.
Future work might look at short video clips instead of still images, since movement adds a lot of information about what someone is feeling.
The fact that combining both sources improves results confirms what we expected: text and faces each reveal different parts of the emotional picture. In practical settings like analyzing recorded video calls or social media posts with embedded images this combination could be even more powerful.
Looking forward, three directions stand out as especially worth pursuing.
First, building smarter fusion systems that can tell which input is more reliable in a given moment and weight it accordingly. Second, developing personalized models that learn to recognize the unique way each person expresses themselves, since emotional expression varies greatly between individuals and cultures. Third, expanding to include other types of input, like the tone of someone's voice or even physiological signals, to make the system even more complete.

6. References

1. E. Batbaatar, M. Li, and K. H. Ryu, 'Semantic-Emotion Neural Network for Emotion Recognition From Text,' IEEE Access, vol. 7, pp. 111866–111879, 2019.
2. D. Kalla, N. Smith, F. Samaah, and K. Polimetla, 'Facial Emotion and Sentiment Detection Using Convolutional Neural Network,' INDJAIR, vol. 1, no. 1, pp. 1–13, 2021.
3. T. Mikolov, K. Chen, G. Corrado, and J. Dean, 'Efficient Estimation of Word Representations in Vector Space,' Proc. ICLR, 2013.

4. J. Pennington, R. Socher, and C. Manning, 'GloVe: Global Vectors for Word Representation,' Proc. EMNLP, 2014.
5. P. Bojanowski et al., 'Enriching Word Vectors with Subword Information,' Trans. ACL, vol. 5, pp. 135–146, 2017.
6. Y. LeCun, Y. Bengio, and G. Hinton, 'Deep Learning,' Nature, vol. 521, pp. 436–444, 2015.
   A. Krizhevsky, I. Sutskever, and G. Hinton, 'ImageNet Classification with Deep CNNs,' Proc. NIPS, 2012.
7. S. Hochreiter and J. Schmidhuber, 'Long Short-Term Memory,' Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
8. K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition,' Proc. CVPR, 2016.
9. R. Cowie et al., 'Emotion Recognition in Human-Computer Interaction,' IEEE Signal Processing Magazine, vol. 18, no. 1, pp. 32–80, 2001.