Image-Based Facial Emotion Recognition: A Deep Learning-Centric Approach with Comparative Analysis Against Traditional ML Techniques

Sujan Khanal
u3258630@uni.canberra.edu.au
Rohit Baral
u3268702@uni.canberra.edu.au

University of Canberra
Faculty of Science and Technology
Canberra, Australia

Abstract

This project focused on developing an image-based facial emotion recognition system using deep learning techniques, with traditional machine learning methods employed for comparative analysis. The primary objective was to accurately classify facial expressions into seven emotion categories while addressing challenges such as class imbalance and feature limitations. Classical models, including k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM), were implemented using handcrafted features like Local Binary Patterns (LBP) and a combination of Histogram of Oriented Gradients (HOG) with LBP, serving as baseline references. The core of the project involved designing and training a Convolutional Neural Network (CNN) architecture, enhanced with data augmentation, focal loss, and class-balanced batch generation to improve performance on underrepresented emotion classes. The CNN-based approach significantly outperformed traditional methods across key evaluation metrics including accuracy, precision, recall, and F1-score, particularly improving recognition of minority classes such as 'disgust' and 'fear'. Additionally, the trained CNN model was integrated into a real-time facial emotion recognition system using webcam input, demonstrating its practical applicability in live scenarios. This project highlighted the superiority of deep learning for facial emotion recognition and validated the use of class-aware strategies and real-time deployment for effective and balanced performance.

1 Introduction

1.1 Background on Facial Emotion Recognition

Facial emotion recognition (FER) has become increasingly significant in advancing human-computer interaction, healthcare diagnostics, education technology, and security systems. It aims to interpret human emotions by analysing facial expressions captured in images or video. Traditional approaches relied on handcrafted features coupled with classical classifiers, but these methods often lacked generalization across variations in pose, lighting, and expression [1]. Deep learning has revolutionized visual recognition [2], particularly through

Convolutional Neural Networks (CNNs), which enable the learning of complex, discriminative features directly from image data. Residual networks further improved the training of deep architectures [3].

1.2 Motivation

Despite progress in deep learning, several challenges remain unresolved. Chief among Despite progress in deep learning, several challenges remain unresolved. Chief among them is class imbalance—where certain emotions such as "disgust" or "fear" are underrepresented in datasets like FER-2013—leading to poor classification performance on minority classes [4]. Additionally, the growing demand for real-time systems necessitates FER solutions that are not only accurate but also computationally feasible in live environments. This project was motivated by the need for an accurate, fair, and deployable FER solution that performs consistently across all classes and works in real-time.

1.3 Problem Statement

Many FER models focus solely on overall accuracy, overlooking the performance disparity across emotional categories. Classical ML models, while lightweight, struggle with feature expressiveness. Deep learning models, though highly capable, can overfit imbalanced data and lack real-time adaptability. Thus, there is a need for a balanced, high-performing, and real-time compatible FER system that improves upon traditional methods.

1.4 Objectives and Scope

- The key objectives of this project were:
- Implement classical ML models (k-NN, SVM) as a base for comparison.
- Develop a CNN model equipped with adaptive training strategies
- Compare ML and CNN models using evaluation metrics.
- Deploy the CNN model in a real-time FER system.

This study focused on grayscale image inputs across seven emotion categories and did not explore multimodal or 3D recognition.

2 Literature Review

Facial emotion recognition (FER) has been studied extensively through both traditional machine learning (ML) techniques and deep learning methods. Early approaches typically relied on handcrafted features such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), Gabor filters, and geometric descriptors. These features were then fed into classifiers like Support Vector Machines (SVM), k-Nearest Neighbours (k-NN), or Random Forests to perform emotion classification. Although effective to an extent, such systems often lacked scalability and struggled with generalisation across different lighting conditions, poses, and subject variability [5].

LBP has been widely adopted due to its ability to capture local texture patterns efficiently. Shan et al. used LBP with SVM to classify facial emotions and demonstrated competitive performance on the FER dataset [5]. HOG, another texture descriptor, was later combined with LBP to improve feature richness for emotion classification, although this often came at

the cost of increased computational complexity.

The emergence of deep learning brought transformative changes to FER. Convolutional Neural Networks (CNNs), in particular, replaced the need for manual feature engineering by automatically learning spatial hierarchies from raw image pixels. Krizhevsky et al. showcased the power of CNNs through their success on the ImageNet challenge, laying the foundation for deep architectures in visual recognition tasks [2]. Later, He et al. introduced Residual Networks (ResNet), which allowed for much deeper models by resolving the vanishing gradient problem, further improving recognition accuracy [3].

However, despite their capabilities, CNN-based FER systems often suffer from class imbalance in datasets like FER-2013, where minority classes such as "disgust" or "fear" are underrepresented. This imbalance leads to biased learning where models overfit to dominant classes. Goodfellow et al. highlighted the importance of addressing representation challenges and introduced techniques such as data augmentation and advanced loss functions to improve fairness and generalisation [4].

Recent advancements in FER focus on integrating augmentation strategies, custom loss functions like focal loss, and class-aware batch generation to enhance performance on underrepresented classes. These methods have shown promise in not only increasing accuracy but also in improving recall for minority emotions in imbalanced datasets.

In summary, the literature indicates a clear progression from handcrafted feature-based ML models to deep learning frameworks that can autonomously learn complex patterns. Yet, overcoming dataset imbalance and deploying models in real-time environments remain key research challenges that this project addresses.

3 Methodology

This project adopted a two-stage methodology:

1. Baseline evaluation using traditional machine learning (ML) approaches with handcrafted features, and
2. Development and training of a deep learning model based on a Convolutional Neural Network (CNN)

The goal was to assess the comparative performance of both approaches on facial emotion recognition tasks, particularly under the constraint of class imbalance, and to deploy a real-time system for practical inference.

3.1 Dataset Description and Exploratory Data Analysis (EDA)

The experiments were conducted on a labelled facial emotion dataset containing seven emotion classes: angry, disgust, fear, happy, sad, surprise, and neutral. The dataset was extracted from the Kaggle website.

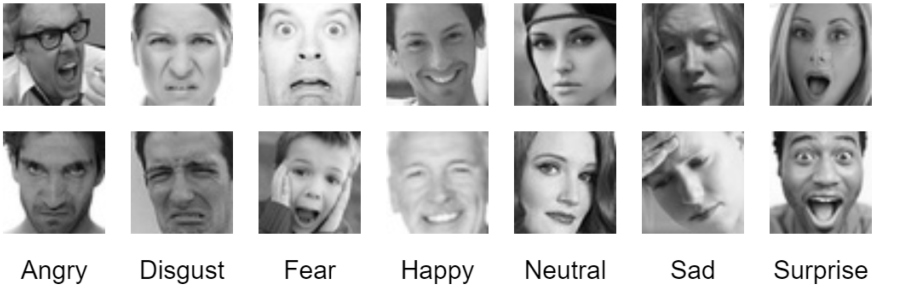


Figure 1: FER-2013 Dataset [7]

The FER−2013 dataset consists of 48x48 pixel grayscale images of faces, where the faces have been automatically registered to ensure that they are centered and occupy a consistent amount of space in each image. However, like many standard FER datasets, it exhibited class imbalance, with certain emotions (e.g., disgust and fear) being underrepresented. This imbalance posed a challenge for traditional classification algorithms and necessitated the use of specialised techniques during deep learning model training [4].

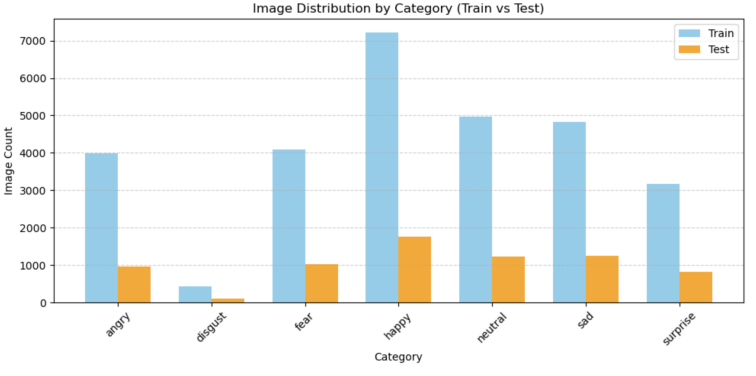


Figure 2: Distribution of Train and Test Data Files

To better understand the distribution, an exploratory data analysis was conducted. The chart shows the number of training and test images per emotion category. The happy class dominated the dataset with over 7,200 training images, while disgust had the fewest, with only 436 training and 111 test images. This imbalance is visually evident in the bar chart and highlights the need for techniques such as data augmentation and class-balanced training strategies. The EDA reconfirmed the necessity of focal loss and resampling methods to ensure that underrepresented classes are appropriately learned by the model.

3.2 Preprocessing and Feature Extraction

This stage prepares raw data for input into ML models by transforming images into numerical features. Feature extraction helps isolate meaningful patterns that distinguish emotions, particularly when using classical approaches that rely on explicit descriptors.

For traditional ML models, feature extraction was performed using the following methods:

- **Local Binary Patterns (LBP):** A texture descriptor that labels the pixels of an image by thresholding the neighbourhood of each pixel and considering the result as a binary number. LBP captures micro-patterns in local regions and is computationally efficient, making it suitable for facial texture analysis [5].
- **Histogram of Oriented Gradients (HOG):** A feature descriptor that counts occurrences of gradient orientation in localised portions of an image. It is effective in capturing edge structure and shape information [6].
- **LBP + HOG Combination:** A hybrid descriptor formed by concatenating LBP and HOG features. This approach enhances the feature representation by combining local texture and gradient orientation information.

All extracted features were normalised using z-score standardisation prior to classification.

3.3 Classical Machine Learning Models

This step evaluates baseline performance using lightweight classifiers. Classical ML models offer interpretability and quick deployment but generally lack the flexibility to capture deep spatial features inherent in facial expressions. Two classifiers were implemented:

- **k-Nearest Neighbours (k-NN):** A non-parametric, instance-based learning algorithm where classification is based on the majority label among the k closest data points in the feature space.
- **Support Vector Machine (SVM):** A supervised learning algorithm that identifies the optimal hyperplane to separate classes. The RBF kernel was used to handle non-linearly separable data. Hyperparameters such as kernel scale and box constraint were tuned using grid search.

These classical models served as baseline references for performance comparison.

3.4 Deep Learning Model

Deep learning models like CNNs eliminate the need for manual feature engineering by learning features directly from data. This section describes the architecture and strategies applied to improve performance and fairness under class imbalance.

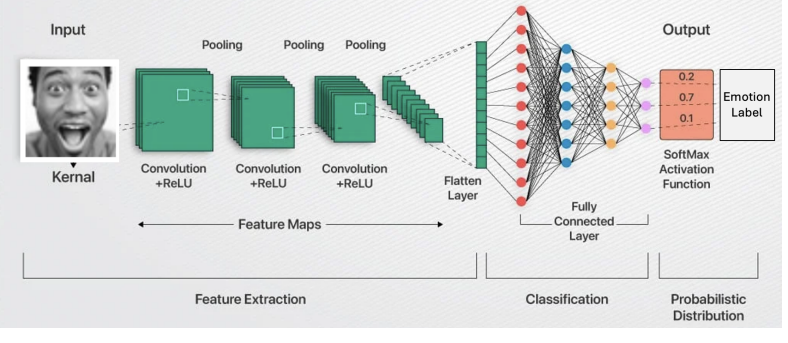


Figure 3: Convolutional Neural Network [8]

The central component of the project was a CNN architecture developed to learn hierarchical representations directly from image pixels. The model consisted of:

- Convolutional layers with ReLU activations
- Batch normalisation to stabilise learning
- Dropout layers to prevent overfitting
- Max pooling for spatial dimensionality reduction
- Fully connected layers ending with a softmax output layer

To enhance model learning under class imbalance, the CNN was equipped with adaptive training strategies, including:

- **Data Augmentation:** A method of synthetically expanding the training dataset by applying transformations like flipping, rotation, zooming, and shifting. It improves generalisation by simulating real-world variation.
- **Focal Loss:** A modified cross-entropy loss function that reduces the relative loss for well-classified examples and focuses learning on hard, misclassified examples. It is particularly effective for addressing class imbalance [9].
- **Class-Balanced Batch Generation:** A technique to ensure that each training batch includes an equal number of samples from each emotion class, improving convergence and fairness during training.

3.5 Model Training and Evaluation

This step involves configuring and monitoring the model's learning process. Evaluation metrics are used to quantify the model's predictive performance, especially in the context of imbalanced classes.

The CNN was trained using the Adam optimiser with learning rate scheduling and early stopping based on validation loss. The following metrics were used for evaluation:

- Accuracy
- Precision, Recall, and F1-score (macro-averaged)
- Confusion matrix for class-wise performance

3.6 Real-Time Integration

The final phase involved deploying the trained model in a real-time system. This validated its usability in live environments, where low-latency prediction and practical face detection were essential.

To validate practical deployment, the trained CNN model was integrated into a real-time facial emotion recognition system using a webcam stream. OpenCV was used to capture frames, apply preprocessing, and pass inputs to the CNN for prediction. Results were displayed live, providing immediate feedback on detected emotions.

4 Results and Evaluation

This section presents the performance outcomes of the models implemented and evaluated in this study. Classical machine learning (ML) models served as baselines, while a CNN-based deep learning model was the primary focus. The evaluation compared these models across several performance metrics to determine their effectiveness in facial emotion recognition (FER), particularly in handling class imbalance.

4.1 Evaluation Strategy

The Evaluation Strategy outlines the methods used to assess model performance. Evaluation is essential to quantify predictive accuracy, highlight class-wise effectiveness, and identify areas for model refinement.

The models were assessed using a stratified 5-fold cross-validation approach to ensure balanced representation of all classes. Key performance metrics used included:

- Accuracy: Proportion of correctly predicted samples across all classes.
- Precision: Ratio of true positives to total predicted positives, indicating correctness of positive predictions.
- Recall: Ratio of true positives to total actual positives, reflecting the model's sensitivity.
- F1-score: Harmonic mean of precision and recall, especially useful for imbalanced class distributions.
- Confusion Matrix: Provided a visual summary of true vs. predicted class performance.

4.2 Performance of Classical ML Models employing Feature Extractors

The results of baseline models, which serve as a reference to measure improvement with deep learning, are discussed in this section. The evaluation included k-Nearest Neighbours (k-NN) classifiers applied to two feature representations: LBP histograms and combined HOG+LBP features.

Classification Report (Best k-NN + LBP Histogram):

	precision	recall	f1-score	support
angry	0.17	0.35	0.22	958
disgust	0.13	0.24	0.17	111
fear	0.17	0.23	0.20	1024
happy	0.28	0.28	0.28	1774
neutral	0.22	0.13	0.16	1233
sad	0.22	0.11	0.14	1247
surprise	0.36	0.19	0.25	831
accuracy			0.22	7178
macro avg	0.22	0.22	0.20	7178
weighted avg	0.24	0.22	0.21	7178

Classification Report: Best KNN with HOG + LBP

	precision	recall	f1-score	support
angry	0.34	0.50	0.40	958
disgust	0.32	0.50	0.39	111
fear	0.39	0.40	0.40	1024
happy	0.57	0.74	0.64	1774
neutral	0.48	0.38	0.42	1233
sad	0.53	0.23	0.32	1247
surprise	0.67	0.58	0.62	831
accuracy			0.49	7178
macro avg	0.47	0.47	0.46	7178
weighted avg	0.50	0.49	0.48	7178

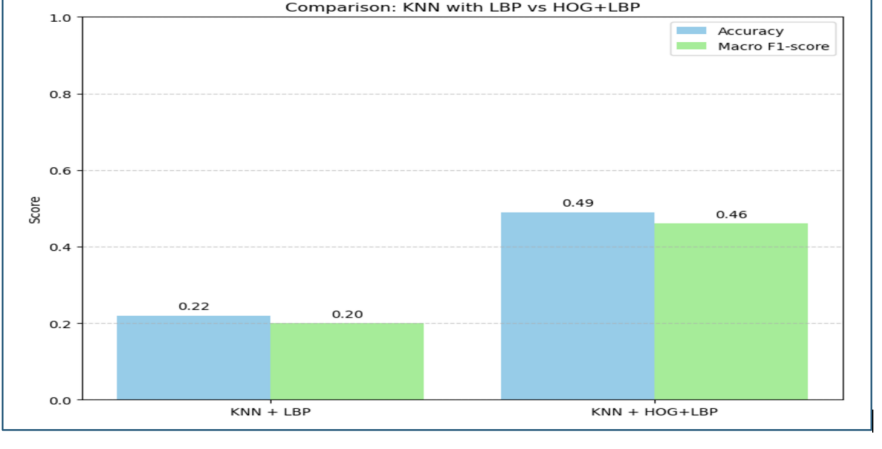


Figure 4: k-NN Feature Extractors Comparison

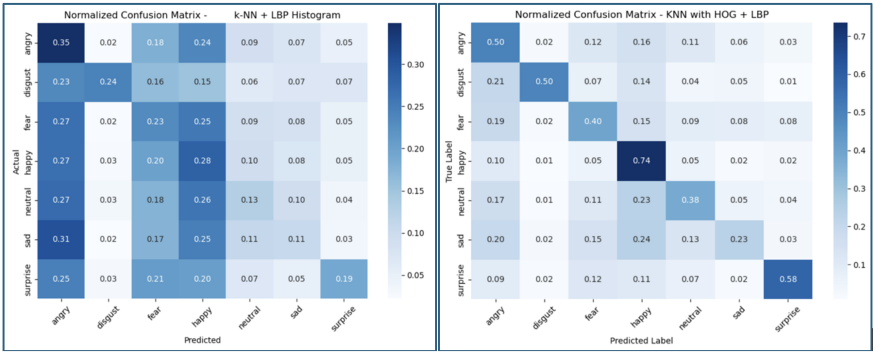


Figure 5: Comparison between Confusion Matrices

k-NN with LBP Histogram: The k-NN model using Local Binary Pattern (LBP) histograms achieved an overall accuracy of 22% and a macro-averaged F1-score of 0.20, as shown in the classification report and bar chart in Figure 4. While LBP provided a compact and efficient texture representation, the model struggled with class imbalance and expression overlap. Minority emotions like disgust and fear yielded low F1-scores of 0.17 and 0.20 respectively, and recall values remained below 0.25. Even common emotions like happy and sad showed weak performance (F1-scores of 0.28 and 0.14 respectively), indicating limited feature expressiveness for high-level facial cues.

The confusion matrix revealed significant misclassifications between similar-looking expressions (e.g., sad and neutral, or angry and fear), consistent with known challenges in LBP-only models. These limitations are reflective of LBP's sensitivity to subtle texture variations but its inability to encode larger structural patterns.

k-NN with HOG + LBP Histogram: When Histogram of Oriented Gradients (HOG) features were combined with LBP, performance improved markedly. The combined model reached an accuracy of 49% and a macro-averaged F1-score of 0.46, more than doubling the performance of the LBP-only model. This boost was observed across nearly all classes:

- Disgust: F1 increased from 0.17 to 0.39
- Fear: F1 improved from 0.20 to 0.40
- Happy: F1 rose significantly from 0.28 to 0.64
- Surprise: F1 increased from 0.25 to 0.62

These improvements can be attributed to the complementary nature of the features: LBP captures fine-grained textures, while HOG captures edge orientation and global facial structure. This combination allowed the model to differentiate expressions better, particularly those involving prominent edge and shape changes (e.g., smiles or wide eyes).

Comparative Insights: Figure 4 and Figure 5 illustrate the gap in performance between the two models. The HOG+LBP combination provided a more expressive feature representation and demonstrated significantly higher discriminative capability, particularly for both majority and minority classes. However, while performance gains were strong, the approach remained non-trainable, heavily reliant on manually engineered features, and limited in generalisation to unseen contexts.

SVM with HOG + LBP Histogram: The Support Vector Machine (SVM) model trained on HOG + LBP features yielded the best performance among all traditional ML approaches tested. It achieved an accuracy of 57%, a macro-averaged F1-score of 0.57, and consistently higher class-wise precision and recall compared to the k-NN variants.
From the classification report, Figure 6:

- Happy, surprise, and disgust categories attained F1-scores of 0.73, 0.74, and 0.64, respectively.
- Even difficult classes like fear, angry, and sad achieved moderate F1-scores around 0.44–0.47, which were substantially better than those achieved by the k-NN model.

The normalised confusion matrix, Figure 6, further confirmed this performance gain, showing less dispersion across off-diagonal values and better clustering of correct predictions (e.g., 78% of happy images and 70% of surprise images correctly classified). The disgust class, despite its minority presence in the dataset, reached 50% recall, indicating SVM's effectiveness in handling rare classes when paired with informative feature sets.

This outcome highlights the advantage of using a margin-based learning algorithm like SVM for high-dimensional, structured feature spaces. The model was better equipped to form non-linear boundaries that captured subtle differences in facial texture and edge orientation, particularly when using the complementary HOG and LBP descriptors.

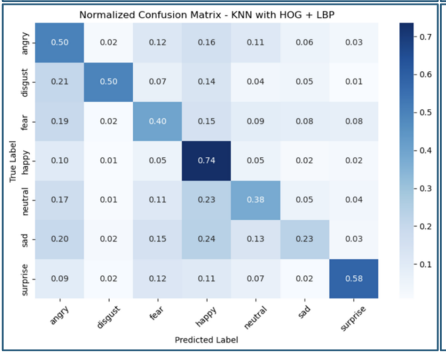
Comparative Insights: k-NN vs SVM (with HOG + LBP Histogram):

Classification Report: Best KNN with HOG + LBP

	precision	recall	f1-score	support
angry	0.34	0.50	0.40	958
disgust	0.32	0.50	0.39	111
fear	0.39	0.40	0.40	1024
happy	0.57	0.74	0.64	1774
neutral	0.48	0.38	0.42	1233
sad	0.53	0.23	0.32	1247
surprise	0.67	0.58	0.62	831
accuracy			0.49	7178
macro avg	0.47	0.47	0.46	7178
weighted avg	0.50	0.49	0.48	7178

Classification Report: SVM with HOG + LBP

	precision	recall	f1-score	support
angry	0.45	0.44	0.44	958
disgust	0.89	0.50	0.64	111
fear	0.47	0.42	0.44	1024
happy	0.69	0.78	0.73	1774
neutral	0.53	0.53	0.53	1233
sad	0.46	0.48	0.47	1247
surprise	0.78	0.70	0.74	831
accuracy			0.57	7178
macro avg	0.61	0.55	0.57	7178
weighted avg	0.57	0.57	0.57	7178



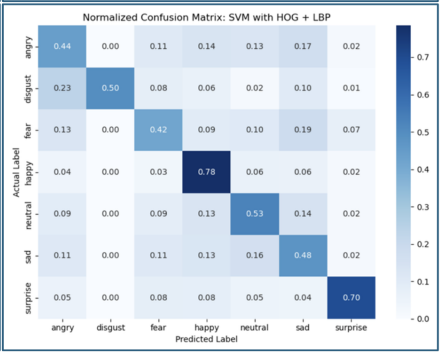


Figure 6: k-NN vs SVM summary

Both models utilised the same composite feature vector combining Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP)—a hybrid descriptor designed to

leverage the texture-sensitivity of LBP and the edge-orientation sensitivity of HOG. However, their performance differed significantly in many aspects.

k-NN is a non-parametric, lazy learner that classifies test instances by looking at the k nearest training samples in feature space. Its performance is highly sensitive to local data density, feature scaling, and noise. It assumes that similar emotions have tightly clustered, compact features, which is often not true in facial emotion datasets due to expression ambiguity and overlap. The concatenation of HOG and LBP results in a high-dimensional vector. While this is beneficial for representing complex facial patterns, it also increases the risk of overfitting or distance distortion in k-NN, due to the "curse of dimensionality."

SVM, in contrast, is a global learner that finds the optimal separating hyperplane with maximum margin between classes. SVM handles high-dimensional features more effectively due to regularisation techniques that prevent overfitting. When coupled with a non-linear kernel (RBF), it transforms the input space to better resolve overlapping distributions. This approach reduces boundary confusion, especially in classes with subtle differences (e.g., fear vs. sad). SVM capitalised on the high-dimensional HOG+LBP representation without being overwhelmed by irrelevant or redundant features. SVM's ability to generalise better across high-dimensional, sparse, and overlapping data led to superior performance.

4.3 Deep-Learning (CNN) Performance

The convolutional neural network (CNN) developed for this facial emotion recognition task followed a structured and technically informed design, integrating multiple techniques to address class imbalance, dataset limitations, and feature extraction challenges. The final model achieved a test accuracy of 63%, with relatively high performance across several emotion categories such as happy (F1-score: 0.84) and surprise (F1-score: 0.77), while minority classes like disgust and fear also saw marked improvement.

4.3.1 Network Architecture and Feature Hierarchy

The CNN was constructed with multiple convolutional blocks that successively expanded the representational power of the network:

- **Convolutional Layers:** These layers use small filters (mostly 3x3) with ReLU activations. Extracted low- and mid-level facial features such as edges, textures, and localised patterns from the input images. Deeper convolutional layers capture mid- and high-level semantic features, such as facial landmarks (eyebrows, mouth curvature, wrinkles). Each layer builds on spatial hierarchies.

- **Batch Normalisation:** Normalised activations after each convolution, and reduced internal covariate shift , allowing for faster convergence and smoother gradients.

- **LeakyReLU Activation:** LeakyReLU was employed as the activation function in this CNN architecture to introduce non-linearity while mitigating the "dying ReLU" problem, where standard ReLU units can become inactive for all inputs. Unlike ReLU, which outputs zero for all negative inputs, LeakyReLU allows a small, non-zero gradient when the input is negative. This enables gradient flow through all neurons, ensuring better convergence and more stable learning, especially important in deeper networks or when training on imbalanced datasets. Its application supported the model in

learning complex feature patterns without discarding potentially useful negative activations.

- **Max Pooling:** Downsampled feature maps to retain essential information while reducing spatial complexity.
- **Dropout Layers:** Applied after convolution or dense layers. Prevented overfitting by randomly deactivating neurons during training. Helped model generalise better to unseen data.
- **Fully Connected Layer and Softmax Output:** Mapped the final high-level feature vectors to class probabilities across the seven emotion categories.

```
# Input layer for grayscale image of shape (IMG_SIZE x IMG_SIZE x 1)
inputs = layers.Input(shape=(IMG_SIZE, IMG_SIZE, 1))

# Initial convolution block
x = layers.Conv2D(64, (3, 3), padding='same')(inputs)   # First conv layer
x = layers.BatchNormalization()(x)                      # Batch normalization
x = layers.LeakyReLU()(x)                               # LeakyReLU activation

# Residual block with 64 filters
x = residual_block(64)(x)
x = layers.MaxPooling2D()(x)                            # Downsampling with max pooling

# Convolution + Residual block with 128 filters
x = layers.Conv2D(128, (3, 3), padding='same')(x)
x = layers.BatchNormalization()(x)
x = layers.LeakyReLU()(x)
x = residual_block(128)(x)
x = layers.MaxPooling2D()(x)

# Convolution + Residual block with 256 filters
x = layers.Conv2D(256, (3, 3), padding='same')(x)
x = layers.BatchNormalization()(x)
x = layers.LeakyReLU()(x)
x = residual_block(256)(x)
x = layers.MaxPooling2D()(x)

# Global average pooling reduces each feature map to a single value
x = layers.GlobalAveragePooling2D()(x)

# Dense layer with 256 units
x = layers.Dense(256)(x)
x = layers.BatchNormalization()(x)
x = layers.LeakyReLU()(x)
x = layers.Dropout(0.5)(x)  # Dropout for regularization

# Output layer with softmax activation for multi-class classification
outputs = layers.Dense(NUM_CLASSES, activation='softmax')(x)
```

Figure 7: CNN Code Architecture

4.3.2 Performance Interpretation - CNN

The performance of CNN can be in-detail explained by interpreting the CNN-Confusion Matrix and CNN-Classification Report.

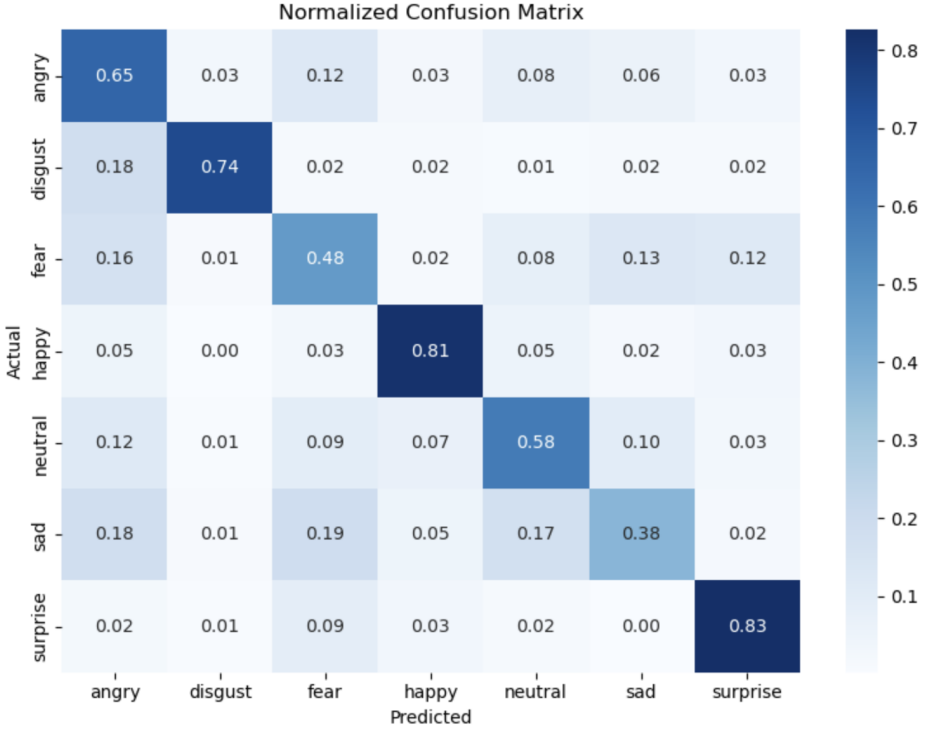


Figure 8: CNN - Confusion Matrix

Final Classification Report:

	precision	recall	f1-score	support
angry	0.48	0.65	0.55	958
disgust	0.53	0.74	0.61	111
fear	0.45	0.48	0.47	1024
happy	0.87	0.81	0.84	1774
neutral	0.60	0.58	0.59	1233
sad	0.57	0.38	0.46	1247
surprise	0.72	0.83	0.77	831
accuracy			0.63	7178
macro avg	0.60	0.64	0.61	7178
weighted avg	0.64	0.63	0.63	7178

Figure 9: CNN - Classification Report

CNN Classification Report Insights:

- **Happy (0.84):** Highest performance due to distinct facial features (e.g., wide smile, eyes).
- **Surprise (0.77):** Features like wide-open eyes/mouth are easily detectable.
- **Disgust (0.61):** Improved by model's ability to capture nose wrinkle, upper lip raise.
- **Sad/Fear/Angry (<0.55):** Lower scores due to high intra-class similarity and inter-class confusion.
- **Sad(0.46):** Most confused with angry or neutral — subtle facial differences.

Confusion matrix shows overlap, e.g., "fear" misclassified as "angry" (16%), "sad" (13%). Facial expressions like sad, angry, and neutral often share overlapping muscle movements, this complexity affected separatability.

Training vs Validation Accuracy - CNN:

- Training accuracy steadily increased, surpassing 78%, while validation accuracy converged to 63%.
- Validation loss plateaued despite continuing training, suggesting minor overfitting.
- The narrow gap between training and validation curves confirmed the impact of augmentation, regularisation, and batch balancing in improving generalisation. Still some room for improvements.

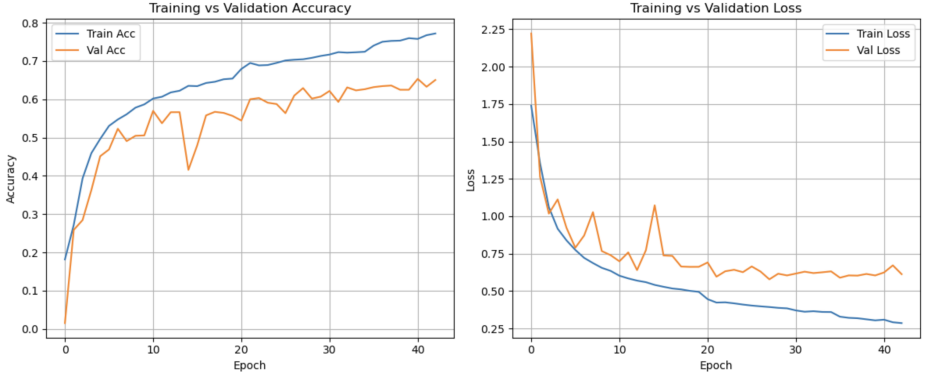


Figure 10: Training vs Validation Plot

4.3.3 Deep Learning-CNN vs ML(k-NN and SVM)

The comparative evaluation between classical machine learning models and the deep learning-based CNN revealed significant performance differences in terms of accuracy, generalisation, and class-specific reliability. While the ML models offered baseline interpretability and quicker training cycles, the CNN outperformed both k-NN and SVM in almost every major performance metric.

The CNN model, trained end-to-end on raw images with automated hierarchical feature extraction, reached a final accuracy of 63% and macro F1-score of 0.61. It demonstrated superior learning capacity for complex, non-linear patterns through multiple convolutional layers. The CNN also incorporated LeakyReLU activation, focal loss, data augmentation, and class-balanced batch generation, all of which enhanced its ability to handle intra-class variability and data imbalance—limitations that classical models were not structurally equipped to overcome.

Furthermore, CNN's confusion matrix showed clearer diagonal dominance, indicating higher true positive rates across all classes. This was particularly evident in classes like happy, surprise, and disgust, where CNN showed significantly higher F1-scores compared to k-NN and SVM, underscoring its advantage in capturing semantic spatial features.

In summary, the CNN model not only surpassed classical ML baselines in accuracy but also provided more balanced and scalable emotion recognition. Its performance reflected the strength of data-driven feature learning over manually crafted descriptors, especially in complex vision tasks like facial expression classification.

4.4 Real-Time Evaluation

To extend the applicability of the CNN model beyond offline inference, a real-time facial emotion recognition system was developed. This system used OpenCV for webcam capture and TensorFlow/Keras to load the trained CNN model. The deployment pipeline incorporated consistent preprocessing steps that mirrored the training phase, ensuring compatibility and performance continuity.

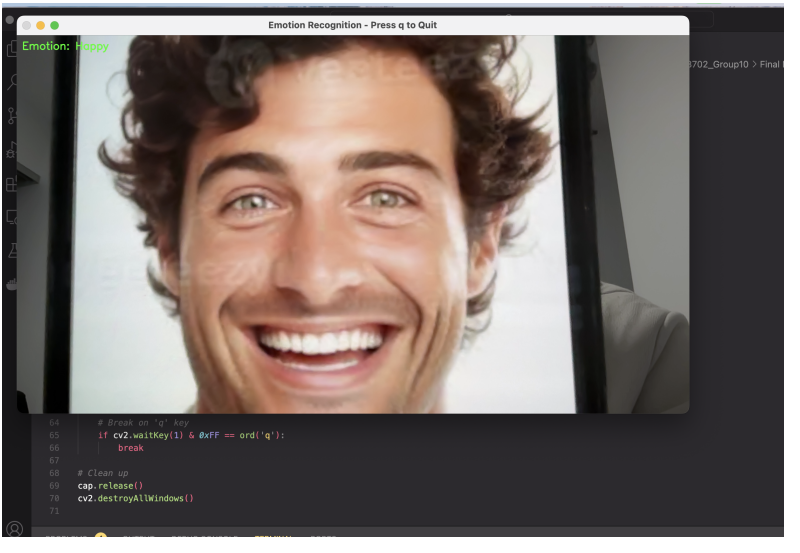


Figure 11: Real Time Facial Emotion Recognition

Each incoming video frame was converted to grayscale and resized to 48×48 pixels, aligning with the input shape of the model. The image was then normalised to the [0,1] range

and reshaped to include both batch and channel dimensions. This preprocessing allowed for consistent input formatting required by the CNN, which had been trained on greyscale FER images.

The system loaded the trained model using a custom-defined focal loss function. This focal loss, previously used during training, was also incorporated during model loading to maintain class imbalance sensitivity. The model then predicted emotion probabilities, and the class with the highest probability was selected as the detected emotion.

For user interaction, the system overlayed the predicted emotion label directly onto the live video feed using Opencv function. This real-time feedback loop allowed the model to continuously analyse emotions with low latency. OpenCV-AVFoundation was used for smooth webcam access on macOS systems.

Limitations of the Real-Time Facial Emotion Recognition System:
While the developed real-time emotion recognition pipeline demonstrated reliable classification under controlled conditions, several practical limitations emerged during deployment:

- **Multiple Faces and Rapid Movements:** The current model assumes a single prominent face per frame. In real-world scenarios involving multiple subjects or sudden head movements, the accuracy can degrade due to motion blur and overlapping facial regions.
- **Complex Backgrounds and Lighting Variation:** The system's performance was sensitive to inconsistent lighting and cluttered backgrounds. Changes in ambient illumination or partial occlusions caused reduced confidence in predictions, highlighting the need for more advanced spatial and temporal filtering techniques.
- **Noise, Motion Blur, and Low Camera Quality:** Real-time prediction was adversely affected by poor webcam resolution, frame noise, and motion blur during fast movements. These factors introduced visual artefacts that the CNN had not seen during training, leading to classification errors.

Future improvements could include integrating face tracking, temporal smoothing, and adapting the model to handle multi-face detection and low-quality inputs using pre-processing pipelines or lightweight denoising networks.

5 Conclusion

This project primarily focused on developing and analysing a deep learning-based solution for facial emotion recognition using convolutional neural networks (CNNs). While classical machine learning models such as k-NN and SVM served as comparative baselines, the CNN emerged as the central and most effective component, demonstrating superior performance in both accuracy and class-wise generalisation. By incorporating focal loss, class-balanced training, and advanced data augmentation, the CNN effectively addressed challenges such as class imbalance and overfitting. Furthermore, its deployment in a real-time setting validated the model's practical applicability. Overall, the project highlighted the effectiveness of deep learning in capturing nuanced facial expressions, affirming their suitability for complex, real-world human–computer interaction tasks.

6 Lessons Learnt and Recommendations

Based on the outcomes and challenges encountered throughout this project, the following recommendations are proposed for future work and enhancements. In parallel, several key lessons were gained during the development process, particularly in deep learning modelling.

Firstly, we observed that deep learning models demand careful architecture design and hyperparameter tuning. The choice of activation functions (e.g., using LeakyReLU), loss functions like focal loss, and batch balancing strategies directly impacted the model's ability to learn from imbalanced data. We also learnt the significant value of data augmentation in improving generalisation, especially when training on constrained or biased datasets.

Secondly, we realised that deep learning models can overfit quickly if not carefully regularised or validated. Monitoring training dynamics using learning curves and incorporating augmentation and dropout were critical in maintaining model stability. The interpretability of CNNs also emerged as a challenge, suggesting the importance of integrating explainable AI techniques in future studies.

Through these insights, we recommend the following for further improvement:

- Adopt transfer learning with powerful pretrained CNN backbones (e.g., ResNet, EfficientNet) to boost performance and reduce training time.
- Explore temporal and attention-based architectures, such as CNN-LSTM or Vision Transformers (ViT), to capture complex emotion transitions.
- Enhance dataset diversity by including more samples from varied demographics and facial variations to improve fairness and generalisation.
- Investigate model compression techniques for lighter deployment without compromising accuracy.
- Incorporate interpretability methods (e.g., Grad-CAM) to understand which facial regions contribute most to each emotion prediction.

References

[1] M. Valstar, T. Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, and M. Pantic, "FERA 2015–Second facial expression recognition and analysis challenge," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Ljubljana, Slovenia, 2015, pp. 1–8.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

[4] I. Goodfellow, D. Erhan, P. Carrier, A. Courville, and Y. Bengio, "Challenges in representation learning: A report on three machine learning contests," *Neural Netw.*, vol. 64, pp. 59–63, Apr. 2015.

[5] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on Local Binary Patterns: A comprehensive study," *Image Vis. Comput.*, vol. 27, no. 6, pp. 803–816, May 2009.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, 2005, pp. 886–893.

[7] C. Białek, A. Matiolański, and M. Grega, "An efficient approach to face emotion recognition with convolutional neural networks," *Electronics*, vol. 12, no. 12, pp. 2707, Jun. 2023. [Online]. Available: https://doi.org/10.3390/electronics12122707

[8] R. Singh, "Decoding CNNs: A Beginner's Guide to Convolutional Neural Networks and Their Applications," *Medium*, 2020. [Online]. Available: https://ravjot03.medium.com/decoding-cnns-a-beginners-guide-to-convolutional-neural-networks-and-their-applications-1a8806cbf536

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, 2017, pp. 2980–2988.