NIB 7088 ARTIFICIAL NEURAL NETWORK COURSEWORK

## Design and Development of Specialized Artificial Neural Networks for Visual Emotion Recognition

# Project Overview

This coursework presents a comprehensive implementation of an Artificial Neural Network (ANN) system designed for visual emotion recognition. The project addresses the critical challenge of accurately identifying human emotions from facial expressions using advanced deep learning techniques, including transfer learning, synthetic data generation, and explainable AI methodologies.

# Question 1: Datasets and EDA (10 marks)

## 1.1 Dataset Preparation and Preprocessing

**Dataset Collection:** The primary dataset used for this project is the FullDataEmoSet, a comprehensive collection of facial expression images organized into distinct emotion categories. The dataset serves as the foundation for training a robust emotion recognition system.

**Raw Dataset Overview:** The raw dataset contains 43,756 readable images distributed across 6 emotion classes:

|  |  |  |
| --- | --- | --- |
| Emotion Class | Count | Percentage |
| Angry | 5,089 | 11.63% |
| Fearful | 4,589 | 10.49% |
| Happy | 13,370 | 30.56% |
| Neutral | 8,268 | 18.90% |
| Sad | 7,504 | 17.15% |
| Surprised | 4,936 | 11.28% |

**Dataset Characteristics:** All images are in grayscale format (mode 'L'), with formats distributed as PNG (64.6%) and JPEG (35.4%). The dataset quality is high with 100% of indexed files loading without errors, no corrupt or unreadable files, and no extreme aspect ratios requiring filtering.

## 1.2 Comprehensive Exploratory Data Analysis (EDA)

**Class Imbalance Analysis:** The dataset exhibits significant class imbalance, with the 'Happy' class containing approximately 2.9 times more samples than the smallest class 'Fearful'. This imbalance presents both challenges and opportunities:

* Statistical measures: Min: 4,589, Max: 13,370, Mean: 7,292.67, Std Dev: 3,336.16
* Quartile distribution shows concentrated data in middle range with outliers
* Implemented class weighting and synthetic data generation to address imbalance

**Image Quality and Format Analysis:** Technical specifications reveal uniform grayscale processing with consistent quality metrics:

* All images converted to grayscale (single channel) for processing efficiency
* No pathological tiny images or extreme distortions detected
* Aspect ratios fall within constrained bands, simplifying augmentation policies
* Area distribution appropriate for uniform 224×224 resizing without artifacts

**Data Preprocessing Pipeline:** A systematic preprocessing pipeline was implemented to ensure data quality and model compatibility:

|  |  |  |
| --- | --- | --- |
| Step | Action | Rationale |
| 1 | Load & verify | Fail fast on corrupt assets |
| 2 | Convert to RGB | Match pretrained backbone expectations |
| 3 | Resize to 224×224 | Standard input size for CNN/ViT variants |
| 4 | Optional center crop/pad | Normalize framing |
| 5 | Augment (train only) | Improve generalization |
| 6 | Normalize (ImageNet mean/std) | Align with pretrained weights |

# Question 2: Solution Design (20 marks)

## 2.1 Artificial Neural Network Architecture Design

**Architecture Selection:** The system employs a transfer learning approach using pretrained convolutional neural networks with custom classification heads. The architecture supports multiple backbone networks:

|  |  |  |
| --- | --- | --- |
| Backbone Type | Identifier | Notes |
| ResNet | resnet50 | Default; custom deeper head (Linear→ReLU→LayerNorm→Dropout→Linear) |
| ConvNeXt | convnext\_base | Via timm (if installed) |
| EfficientNet | tf\_efficientnet\_b3\_ns | Noisy student pretrained weights |
| Vision Transformer | vit\_base\_patch16\_224 | Self-attention interpretability (optional) |

**Progressive Training Strategy:** The training employs a two-stage progressive fine-tuning approach:

|  |  |  |  |
| --- | --- | --- | --- |
| Stage | Epoch Span | Action | Purpose |
| 1 (Warmup) | freeze\_backbone\_epochs (default 3) | Freeze all backbone layers; train classification head | Stabilize new head; prevent abrupt weight drift |
| 2 (Fine-tune) | Remainder | Unfreeze all layers; apply discriminative LR (backbone < head) | Adapt high-level features while preserving general representations |

## 2.2 Approach Comparison and Justification

**Why Artificial Neural Networks are Necessary:** Visual emotion recognition presents unique challenges that necessitate deep learning approaches:

* High-dimensional input space: Facial images contain thousands of pixels with complex spatial relationships
* Hierarchical feature extraction: Emotions manifest through subtle combinations of facial muscle movements
* Non-linear decision boundaries: Emotional expressions involve complex interactions between facial features
* Invariance requirements: Recognition must be robust to lighting, pose, age, and individual differences
* Contextual understanding: Emotions often require understanding of spatial relationships between facial features

**Limitations of Classical Machine Learning:** Traditional machine learning techniques are inadequate for this problem:

* Feature Engineering Bottleneck: Manual feature extraction (HOG, LBP, Gabor filters) requires domain expertise and fails to capture complex patterns
* Limited Representational Capacity: Linear and shallow non-linear models cannot model the complex mappings between raw pixels and emotions
* Poor Scalability: Traditional methods struggle with high-dimensional input spaces and large datasets
* Lack of Hierarchical Learning: Cannot automatically learn low-level edges to high-level semantic features
* Sensitivity to Variations: Classical methods are brittle to pose, lighting, and expression intensity variations

**Why Transfer Learning is Ideal:** The selected transfer learning approach is optimal for emotion recognition:

* Leverages Pre-trained Features: ImageNet-trained CNNs have learned generalizable low-level visual features
* Mitigates Overfitting: Reduces risk of overfitting on limited emotion data by starting with robust representations
* Faster Convergence: Significantly reduces training time compared to training from scratch
* Better Performance: Consistently achieves superior results compared to training CNNs from random initialization
* Resource Efficiency: Requires less computational resources and training data
* Catastrophic Forgetting Prevention: Progressive fine-tuning preserves useful pre-trained representations

# Question 3: Model Development and Evaluation (20 marks)

## 3.1 Model Development and Optimization Process

**Baseline Model Development:** The initial baseline model employed a standard ResNet-50 backbone with frozen weights and a simple linear classification head. This baseline achieved moderate performance but suffered from:

* Limited feature adaptation to emotion-specific patterns
* Suboptimal handling of class imbalance
* Insufficient regularization leading to overfitting

**Iterative Improvements:** The model was systematically improved through multiple iterations:

* Enhanced Classification Head: Replaced simple linear layer with deeper head (Linear→ReLU→LayerNorm→Dropout→Linear)
* Progressive Fine-tuning: Implemented two-stage training with backbone freezing followed by discriminative learning rates
* Advanced Regularization: Added label smoothing (0.05), MixUp augmentation (α=0.4), and Exponential Moving Averages (EMA)
* Class Imbalance Mitigation: Implemented class weighting and synthetic data generation
* Optimization Enhancements: Added gradient clipping (norm=1.0) and cosine annealing learning rate schedule

## 3.2 In-depth Study of Optimization Techniques

**Comprehensive Optimization Analysis:** Multiple optimization techniques were evaluated for their effectiveness in emotion recognition:

|  |  |  |  |
| --- | --- | --- | --- |
| Technique | Strengths | Weaknesses | Application |
| Adam Optimizer | Adaptive learning rates, momentum, sparse gradients handling | Can overshoot minima, memory intensive | Primary optimizer with discriminative LRs |
| Learning Rate Scheduling | Prevents overshooting, improves convergence | Requires tuning, can get stuck in local minima | Cosine annealing with linear warmup |
| Gradient Clipping | Prevents gradient explosion, stabilizes training | May limit learning capacity if too restrictive | Norm clipping at 1.0 |
| Label Smoothing | Improves calibration, reduces overconfidence | May slow initial learning | Applied at 0.05 smoothing factor |
| MixUp Augmentation | Improves generalization, regularization | Can create unrealistic samples | Alpha=0.4 for balanced mixing |
| EMA Weights | Smoother convergence, better generalization | Additional memory overhead | Decay=0.999 for stable averaging |

**Hyperparameter Tuning Critical Analysis:** Systematic hyperparameter optimization significantly improved model performance:

|  |  |  |
| --- | --- | --- |
| Parameter | Final Value | Impact on Performance |
| Learning Rate (Backbone) | 1e-4 | Conservative adaptation prevents catastrophic forgetting |
| Learning Rate (Head) | 1e-3 | Faster convergence of randomly initialized layers |
| Batch Size | 32 | Optimal balance between stability and VRAM usage |
| Weight Decay | 1e-4 | Effective regularization without over-penalization |
| Freeze Epochs | 3 | Sufficient head stabilization without limiting adaptation |
| Early Stopping Patience | 5 | Prevents overfitting while allowing convergence |

# Question 4: Web Application Implementation (10 marks)

**FastAPI Web Application Architecture:** A comprehensive real-time web application was developed using FastAPI framework, providing both REST API endpoints and web interface for emotion recognition:

* RESTful API with /predict endpoint for real-time emotion classification
* CORS middleware enabling cross-origin requests for web integration
* Base64 image processing pipeline with automatic face detection and cropping
* Real-time inference using optimized ONNX model for faster prediction
* Health check endpoint for system monitoring and deployment readiness
* Static file serving for web interface assets and visualizations

**System Testing and User Experience:** Comprehensive testing ensures system reliability and user-friendliness:

* Unit Testing: Individual components tested for correctness and error handling
* Integration Testing: End-to-end pipeline validation from image input to emotion prediction
* Performance Testing: Response time optimization ensuring sub-second inference
* Usability Testing: Interface design validation for intuitive user interaction
* Error Handling: Robust exception management with informative error messages
* Cross-browser Compatibility: Testing across different browsers and devices

**Mobile Application Integration:** A Flutter mobile application extends the platform's reach with native mobile capabilities:

* Cross-platform compatibility (iOS and Android)
* Real-time camera integration for live emotion detection
* ONNX model integration for on-device inference
* Enhanced emotion detection service with face detection pipeline
* Offline capability reducing dependency on network connectivity

# Question 5: Explainable AI (20 marks)

## 5.1 Explainable AI Methodologies Comparison

**Comprehensive XAI Framework:** Four distinct explainable AI methodologies were implemented and compared for emotion recognition interpretability:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Core Principle | Strengths | Limitations | Best Use Case |
| Grad-CAM | Class-specific gradient weights pooled over spatial feature maps | Fast, intuitive heatmaps | Resolution limited to last conv layer | Quick sanity check of focus regions |
| Grad-CAM++ | Higher-order gradient terms for better separation | Finer localization, multi-instance handling | Slightly slower, still layer-resolution bound | Distinguish subtle facial micro-regions |
| SHAP | Shapley value approximation of feature contribution | Theoretically grounded, consistent attribution | Computationally expensive, sensitive to background | Auditing attribution consistency |
| LIME | Local linear surrogate on perturbed superpixels | Model-agnostic, interpretable masks | Instability, segmentation dependency | Explaining individual misclassifications |

## 5.2 Explainable AI Implementation and User Interface

**Implementation Architecture:** The XAI system integrates seamlessly with the emotion recognition pipeline:

* Modular Design: Each XAI method implemented as independent modules with standardized interfaces
* Real-time Generation: Explanations generated on-demand during prediction requests
* Multi-method Comparison: Side-by-side visualization of different explanation methods
* Target Layer Optimization: Automatic selection of optimal layers for gradient-based methods
* Background Sample Management: Intelligent background selection for SHAP explanations

**User Interface and Experience Design:** The XAI interface prioritizes transparency and intuitive interpretation:

* Interactive Heatmap Overlays: Clickable saliency maps with intensity scaling
* Method Comparison Dashboard: Side-by-side visualization of different XAI outputs
* Confidence Scoring: Visual indicators showing prediction confidence and explanation reliability
* Feature Attribution Ranking: Ordered lists of most influential image regions
* Export Functionality: Ability to save and share explanation visualizations
* Responsive Design: Optimized for both desktop and mobile viewing

**Quality Assessment Framework:** Systematic evaluation ensures explanation quality and reliability:

|  |  |  |
| --- | --- | --- |
| Quality Check | Evaluation Method | Pass Criteria |
| Saliency Localization | Compare heatmaps across emotion classes | Emotion-relevant facial regions dominate (eyes, mouth, brows) |
| Attribution Sparsity | Analyze SHAP value distributions | Heavy tail with clear top regions; minimal background dominance |
| LIME Stability | Multiple runs with same seed | Near-identical superpixel importance ordering |
| Class Contrast | Cross-class Grad-CAM comparison | Distinct patterns for different emotions |
| Artifact Detection | Background highlight analysis | Minimal non-face emphasis |

# Question 6: GenAI (20 marks)

## 6.1 Transformer Architecture and Large Language Models

**Transformer Architecture Fundamentals:** The Transformer architecture revolutionized natural language processing and forms the foundation of modern LLMs:

* Self-Attention Mechanism: Enables models to weigh the importance of different tokens in a sequence, capturing long-range dependencies
* Multi-Head Attention: Parallel attention mechanisms allowing the model to focus on different types of relationships simultaneously
* Position Encoding: Injects positional information into the model since attention mechanisms are permutation-invariant
* Feed-Forward Networks: Point-wise fully connected layers that process the attention outputs
* Layer Normalization: Stabilizes training and improves gradient flow throughout deep networks
* Residual Connections: Enable training of very deep networks by mitigating vanishing gradient problems

**Large Language Model Internal Workings:** LLMs like GPT utilize the Transformer decoder architecture with specific adaptations:

* Autoregressive Generation: Models generate text token by token, using previous tokens to predict the next
* Causal Masking: Self-attention is masked to prevent the model from attending to future tokens during training
* Massive Scale: Billions of parameters enable complex pattern recognition and generation capabilities
* Emergent Abilities: Complex behaviors emerge from simple next-token prediction training
* Context Window: Models can process and generate text based on extensive context (thousands of tokens)
* Fine-tuning Capabilities: Pre-trained models can be adapted for specific tasks with additional training

## 6.2 GenAI Implementation for Emotion Recognition

**Synthetic Data Generation Pipeline:** Advanced GenerativeAI techniques were employed to augment the emotion dataset using Stable Diffusion:

* Model Selection: Utilized Stable Diffusion v1.5 for high-quality, controllable image synthesis
* Prompt Engineering: Developed emotion-specific prompts with systematic diversity suffixes
* Quality Control Pipeline: Multi-stage filtering including face detection, blur assessment, and duplicate rejection
* Balanced Generation: Targeted synthesis to address class imbalance in the original dataset
* Metadata Tracking: Comprehensive provenance tracking for generated images
* Integration Pipeline: Seamless integration of synthetic data with original training pipeline

**Generation Strategy Details:** The synthetic data generation employed sophisticated techniques:

|  |  |  |
| --- | --- | --- |
| Aspect | Configuration | Purpose |
| Target Images | 83,000 total | Significant dataset expansion for improved training |
| Inference Steps | 30 | Balance between quality and computational efficiency |
| Guidance Scale | 7.5 | Optimal prompt adherence without over-conditioning |
| Quality Filters | Face detection, blur threshold, deduplication | Ensure high-quality synthetic samples |
| Synthetic Cap | 60% per class in training | Prevent synthetic data dominance |
| Reproducibility | Fixed seed (20250924) | Deterministic generation for reproducibility |

## 6.3 Performance Analysis: GenAI vs Traditional Neural Networks

**Comparative Performance Analysis:** Comprehensive evaluation reveals distinct advantages and trade-offs between GenAI and traditional approaches:

|  |  |  |  |
| --- | --- | --- | --- |
| Aspect | Traditional ANNs | GenAI Models | Winner |
| Training Speed | Fast (hours) | Slow (days for large models) | Traditional |
| Data Requirements | Large labeled datasets | Can leverage unlabeled data | GenAI |
| Inference Speed | Very fast (milliseconds) | Slower (seconds) | Traditional |
| Memory Requirements | Moderate (GBs) | Large (hundreds of GBs) | Traditional |
| Interpretability | Good (with XAI tools) | Limited (black box) | Traditional |
| Generalization | Task-specific | Broad, multi-domain | GenAI |
| Customization | Easy fine-tuning | Complex prompt engineering | Traditional |
| Novel Solution Generation | Limited to training data | Creative, novel outputs | GenAI |

**Innovation Opportunities with GenAI:** GenAI models offer unique opportunities for enhancing emotion recognition systems:

* Data Augmentation at Scale: Generate unlimited diverse training samples addressing any class imbalance
* Zero-Shot Emotion Recognition: Use vision-language models for emotion classification without training data
* Multimodal Integration: Combine visual, textual, and audio modalities for comprehensive emotion understanding
* Personalized Emotion Models: Generate person-specific emotion variants for improved individual recognition
* Real-time Data Synthesis: Generate training data on-demand based on deployment environment feedback
* Bias Mitigation: Generate balanced datasets across demographics to reduce model bias
* Cross-Cultural Adaptation: Generate culture-specific emotional expressions for global deployment
* Adversarial Robustness: Generate adversarial examples to improve model robustness

**Hybrid Approach Benefits:** The optimal solution combines both approaches leveraging their respective strengths:

* GenAI for Data Generation + Traditional ANNs for Classification: Best of both worlds
* Cost-Effective Training: Use GenAI to augment small datasets, then train efficient traditional models
* Rapid Prototyping: GenAI for quick concept validation, traditional models for production deployment
* Continuous Improvement: GenAI generates new training data based on deployment feedback

# Conclusion

This comprehensive coursework successfully demonstrates the design and implementation of a specialized Artificial Neural Network system for visual emotion recognition. The project addresses all key aspects of modern AI development, from robust data preprocessing and architecture design to advanced optimization techniques and explainable AI implementation.  
  
The solution leverages cutting-edge transfer learning approaches with progressive fine-tuning, comprehensive explainability through multiple XAI methodologies, and innovative synthetic data generation using GenAI models. The resulting system achieves high performance while maintaining transparency and interpretability crucial for real-world deployment.  
  
The integration of web and mobile applications ensures practical usability, while the systematic comparison of traditional neural networks with GenAI approaches provides valuable insights for future developments. This work establishes a solid foundation for industry-specific AI applications with emphasis on reliability, explainability, and scalability.