AAI-540 ML Design Document

Team Info

Project Team Group #: 5
Authors: Arifa Kokab
Business Name: The Neurobusiness Corporation
Publication Date: 06/26/2025

Team Workflows

GitHub Project Link: https://github.com/arifakokab/FacialExpressionAnalysisModel

Project Scope

Project Background:

The problem I am addressing is the need for automated, scalable, and objective measurement of human emotional response in neuromarketing and consumer research. Traditional emotion scoring relies on manual analysis, which is slow, subjective, and difficult to scale. My goal is to leverage deep learning to automatically detect and classify emotions from facial expressions in video content, providing marketers and UX researchers with actionable insights at frame-level granularity. The core objective of this model is to process large volumes of video and deliver accurate per-frame emotion predictions for seven core emotions: anger, disgust, fear, happiness, neutral, sadness, and surprise. This is a supervised multiclass image classification problem, utilizing state-of-the-art convolutional neural networks for robust emotion recognition in naturalistic settings.

The impact of this project will be measured by the system's ability to deliver accurate, per-frame emotion predictions (targeting ≥89% validation accuracy and a macro F1 score above 0.8), as well as by the scalability, reproducibility, and auditability of the entire inference and monitoring pipeline as deployed in AWS SageMaker.

Technical Background:

To solve this problem, I adopted a MobileNetV2 convolutional neural network architecture, well-suited for efficient and accurate emotion recognition tasks. The system is evaluated using standard multiclass metrics, including accuracy, macro F1-score, and confusion matrices, as these provide a comprehensive view of both overall and per-class performance. The data sources for this project are the FERPlus and RAF-DB datasets, both widely used benchmarks for facial expression analysis, with a combined total of approximately 34,000 annotated facial images covering all seven target emotions.

The primary challenge was ensuring data quality and balance, as some emotion classes are less represented than others, which could introduce bias. I addressed this by applying stratified sampling and data augmentation techniques to enhance minority classes. The data exploration phase included statistical analysis of class distributions and visualization of sample images to ensure label integrity. My hypothesis was that facial geometry, texture, and key point features—captured through the convolutional filters of the MobileNetV2 architecture—would be the most predictive for emotion classification. I used Google Colab with an A100 GPU for model training, then exported the trained artifact for deployment in AWS SageMaker.

Goals vs Non-Goals:

Goals:

- *The primary goal of this project is to develop and validate a robust deep learning pipeline that can accurately recognize emotions from video frames using batch inference.*
- *The system is designed to support large-scale neuromarketing analytics by enabling the processing of extensive video datasets efficiently.*
- *Integrated monitoring and reproducibility are core objectives, ensuring that every component of the pipeline can be audited and repeated as needed.*
- *The implementation is focused on delivering scalable, cloud-based processing using AWS SageMaker and S3 as the primary infrastructure components.*

Non-Goals:

- *This project does not aim to build a real-time inference endpoint for live emotion recognition.*
- *Extending the model through cross-domain transfer learning with datasets outside of FERPlus and RAF-DB is not within the current scope.*
- *The pipeline does not address non-visual modalities of emotion, such as audio signals or physiological measurements.*
- *Support for mobile or on-device deployment is explicitly out of scope.*
- *The solution does not seek to enable real-time or edge-based applications, as the focus is strictly on cloud-based batch processing and evaluation.*

Solution Overview

My ML system consists of several key components. Video data is uploaded to an AWS S3 bucket, and frames are extracted at a defined interval (e.g., 1 frame per second) using an OpenCV-based pipeline, either on Google Colab or a SageMaker Notebook Instance. Extracted frames are stored in S3 for further processing. The core model—a MobileNetV2 CNN trained on FERPlus and RAF-DB—is registered in SageMaker's Model Registry, ensuring version control and reproducibility. Batch inference is performed using SageMaker Batch Transform, which processes large sets of input frames and writes predictions (as .out files) back to S3. Model monitoring is integrated via AWS CloudWatch, allowing me to track job progress, diagnose errors *such as timeouts, and audit inference runs. Post-processing includes parsing prediction outputs, generating classification metrics, and visualizing per-frame emotions. The system is governed by a notebook-based CI/CD workflow, which automates model registration, packaging, and deployment.*

System architecture is visually depicted in my slides and presentation, showing the flow from video upload, frame extraction, batch inference, monitoring/logging, to evaluation and retraining feedback. Before each new model release, I run evaluation and bias checks on recent inference outputs to ensure sustained performance and fairness.

Data Sources:

The data for this project comes from the FERPlus and RAF-DB datasets, both obtained from Kaggle. FERPlus contains roughly 25,000 facial images, each labeled with one of seven core emotion classes. RAF-DB provides around 9,000 training and 3,000 validation images, similarly annotated. By combining both datasets, I have access to a diverse corpus of about 34,000 images, which improves model generalization and robustness. I selected these datasets due to their prevalence in academic benchmarks and their annotation quality. The main risks associated with these data sources include potential class imbalance (e.g., minority emotions such as disgust) and inherent bias due to limited demographic diversity in the original datasets. No sensitive personal identifiers are present, but as with all facial data, privacy considerations are taken seriously.

Data Engineering:

All image data is stored and organized in AWS S3 buckets, separated into raw input (videos), processed frames, and batch inference outputs. Pre-processing involves converting video to image frames, resizing to 224x224 pixels, normalizing pixel values, and applying data augmentation such as horizontal flipping and color jittering during training. Before input into the ML system, the data is stratified and shuffled to ensure class balance and prevent overfitting. Processed frames are uploaded to S3 for batch inference.

Training Data:

The combined dataset is split into training, validation, and test sets using an 80/10/10 ratio. Stratified sampling ensures that each split contains a representative distribution of all emotion classes. Since both datasets come with high-quality labels, no additional manual labeling was required, but data integrity was checked through visualization and exploratory analysis.

Feature Engineering:

For this computer vision task, raw pixel data serves as the primary input. All images are resized to a consistent dimension and normalized according to the ImageNet mean and standard deviation. No manual feature extraction is performed, as the convolutional layers of MobileNetV2 are designed to learn hierarchical features directly from the data. However, data augmentation *techniques—including random horizontal flips and color jitter—are applied during training to improve robustness. No fields are bucketed or excluded, but only frontal, well-lit face images are used to minimize noise.*

Model Training & Evaluation:

Model training is performed on Google Colab with an A100 GPU, using the MobileNetV2 architecture pretrained on ImageNet. Hyperparameters include a batch size of 64, learning rate of 0.0005, AdamW optimizer, and up to 15 epochs with early stopping based on validation accuracy. The loss function is cross-entropy with label smoothing to mitigate class imbalance. Evaluation is conducted using accuracy, macro F1-score, and confusion matrices, both during training and after batch inference on unseen frames. The model achieved 89.6% validation accuracy and a macro F1 score of 0.831, indicating strong generalization across all emotion classes.

Model Deployment:

After training, the model is packaged (including weights, inference script, and requirements) and uploaded to S3. It is then registered in the SageMaker Model Registry for versioning and governance. I deploy the model as a batch inference pipeline using SageMaker Batch Transform, rather than a real-time endpoint. This decision is based on the use case: I need to process large numbers of video frames asynchronously, and batch transform provides greater scalability and cost efficiency for this workflow. Inference outputs are saved as .out files in S3 for downstream analysis. The instance size used is ml.m5.large.

Model Monitoring:

Model inference jobs are monitored using AWS CloudWatch, which records logs, job status, and errors (such as timeouts or failed invocations). This allows me to audit the batch inference process, debug issues, and ensure system reliability. I also monitor model performance by parsing prediction outputs and generating evaluation metrics after each batch run. If accuracy or fairness metrics degrade, I can trigger a retraining loop and deploy an updated model. Infrastructure monitoring is handled through SageMaker and AWS resource dashboards, tracking resource utilization and job status. No automated data drift monitoring is used at this stage, but periodic evaluation is conducted on fresh data.

Model CI/CD:

Continuous integration and deployment is implemented via notebook-based automation. I use Python scripts to register new model versions, create model package groups and model cards, and automate batch transform jobs in SageMaker. The pipeline includes checkpoints for model artifact validation, registration, deployment, and monitoring. While a fully automated GitHub Actions workflow is possible, the current system relies on reproducible notebooks and clear versioning via the SageMaker Model Registry. Tests include model artifact integrity checks, successful registration, and verification of inference outputs.

Security Checklist, Privacy and Other Risks:

This system does ***not*** *store or process Personal Health Information (PHI), Personally Identifiable Information (PII), or credit card data. No user behavior is tracked or stored. All data is stored in private AWS S3 buckets under my account, with access limited to authorized users. The main privacy consideration is the use of facial images, which, although publicly sourced and anonymized, must still be handled responsibly. There are risks of data or model bias, particularly with underrepresented demographic groups in FERPlus and RAF-DB, which may impact model fairness in deployment. Potential ethical concerns include the misuse of emotion recognition technology, such as non-consensual surveillance or manipulation in advertising. These risks are documented and mitigated by transparent reporting and compliance with data protection standards.*

Future Enhancements:

If I had additional time and resources, I would first expand the training data to include more diverse demographic groups and spontaneous emotion samples, reducing bias and improving generalization. Second, I would implement real-time inference endpoints for interactive applications, possibly leveraging SageMaker Endpoints. Third, I would add automated model and data drift detection, as well as more comprehensive bias and explainability reports, to further improve monitoring and compliance. Finally, I would consider integrating the system with a feature store for more advanced tabular or multi-modal analytics in future iterations.