Project Proposal: Pain Intensity Detection System Using Facial Images

1. Introduction
Chronic and acute pain assessment often relies on subjective reporting. Automating pain-level detection via facial expressions can enhance clinical monitoring and enable timely intervention. This project proposes a complete system—from data-driven model training to an interactive frontend—that classifies pain intensity from adult facial images and provides spoken feedback.

2. Problem Statement
Current pain-assessment methods are labor-intensive and prone to bias. A vision-based approach using static facial images can standardize measurement, reduce workload, and support non-verbal or non-cooperative patients.

3. Objectives

- Develop a multi-class classification model to predict four pain-intensity levels (e.g., none, mild, moderate, severe) from facial images.
- Integrate the trained model into a user-friendly application with two input modes: image upload and real-time camera capture.
- Implement a text-to-speech (TTS) module that announces the patient's pain level and recommends medication when necessary.
- Optimize for highest achievable accuracy and real-time performance.

4. Dataset

- **UNBC–McMaster Shoulder Pain Expression Archive** static frames: 48,398 adult face crops with PSPI scores (0–15).
- **Class Binning**: 0 = no pain; 1–5 = mild; 6–10 = moderate; 11–15 = severe.
- **Preprocessing**: face alignment, normalization, and augmentation (random crop, horizontal flip, color jitter).

5. Methodology
5.1 **Data Preparation**

- Use OpenCV/MTCNN for face detection and alignment.
- Normalize pixel values and apply data augmentation to reduce overfitting.
- Split into train/validation/test sets (70/15/15), stratified by pain class.

5.2 **Model Architecture**

- **Base**: EfficientNet-B4 pretrained on ImageNet (strong feature extractor) or ResNet-50 for comparison.
- **Modification**: Remove original classifier, add dropout (p=0.5), concatenate a 256-d feature vector, and a final dense layer with softmax over four classes.
- **Training**: cross-entropy loss, Adam optimizer (lr=1e-4 with cosine annealing), batch size = 32, early stopping on validation loss.

5.3 **Benchmarking & Accuracy Optimization**

- Experiment with EfficientNet-B4 vs. ResNet-50 vs. InceptionV3.
- Use 5-fold cross-validation to estimate generalization.
- Track metrics: accuracy, F1-score per class, confusion matrix.
- Select best-performing architecture for deployment.

6. Frontend Design

6.1 **Technology Stack**

- **Framework**: Html CSS and js.
- **Camera Integration**: WebRTC + TensorFlow.js/MediaPipe Face Detection for real-time capture.
- **Image Upload**: HTML5 file input with drag-and-drop support.

6.2 **User Interface**

- Two tabs: “Upload Image” and “Live Camera.”
- Display: original image or video feed with bounding box and predicted pain level overlay.

7. Speech Interface

- Integrate a lightweight TTS library (e.g., browser's Web Speech API or Python’s pyttsx3).
- Upon prediction, speak: “Detected pain level: Moderate. Medication recommended.”

8. Evaluation Metrics

- **Classification**: overall accuracy, per-class precision/recall/F1, ROC-AUC curves.
- **Latency**: inference time per image (target <100 ms) and end-to-end response (<500 ms).

9. Project Timeline

Week	**Milestone**
1	Dataset acquisition, pre-processing pipeline
2	Base model implementation and initial training
3	Architecture benchmarking and hyperparameter tuning
4	Final model selection
5	Frontend prototype (upload + live camera)
6	TTS integration and UI polishing
7	System integration testing, performance tuning
8	Documentation, deployment, and final report

10. Resources & Budget

- **Hardware**: NVIDIA GPU (e.g., RTX 3060+), development PC.
- **Software**: Python, PyTorch, OpenCV, Html CSS and JS, Web Speech API.
- **Team**: ML engineer, frontend developer.

11. Deliverables

- Complete end to end project with detailed report.

12. Conclusion

This system will provide an objective, real-time assessment of pain intensity from facial images, improving patient monitoring and enabling rapid clinical responses. The proposed methodology balances state-of-the-art deep learning with practical deployment considerations, ensuring high accuracy and usability.