High Impact Skills Development Program for Gilgit Baltistan

Computer Vision Module Project

Project Title: Expression Classification from Facial Images

Learning Objectives:

- Understand the fundamentals of using a deep ConvNet
- Learn how to use Tensorflow/Keras
- Learn how to use Google Colab
- Learn how to train (fine-tune) a convolutional neural network architecture

Overview:

Learning discriminative features for expressions from facial images captured in the wild is a non-trivial task due to intra-class variations and inter-class similarities. Furthermore, background clutter, illumination changes, large pose variations, and partial or full occlusions make it more challenging. The goal of this project is to design and develop a computer vision system that can classify facial expressions. Such a system can have several real-world applications. For example, expressions can be useful for Human-Computer Interaction based recommendation systems to determine whether to push product information or not. It can also be used to determine the psychological state of a person during online interviews. Overall, this project will provide students with a practical opportunity to implement the concepts learnt during the module. It will also allow them to understand practical problems where computer vision can provide solutions. Students will engage in the identification of tools and techniques that can be employed to solve those problems.

Dataset:

The dataset selected for this task is the **Expression in-the-Wild (ExpW)**[1] dataset. The dataset can be downloaded from the following link:

https://drive.google.com/drive/folders/1SDcI273EPKzzZCPSfYQs4alqjL01Kybq

The **Expression in-the-Wild (ExpW)** dataset is for facial expression recognition and contains 91,793 faces manually labeled with expressions (Figure 1). Each of the face images is annotated as one of the seven basic expression categories: “angry (0)”, “disgust (1)”, “fear (2)”, “happy (3)”, “sad (4)”, “surprise (5)”, or “neutral (6)”.

[1] Zhang, Z., Luo, P., Loy, C. C., & Tang, X. (2018). From facial expression recognition to interpersonal relation prediction. *International Journal of Computer Vision*, *126*, 550-569.

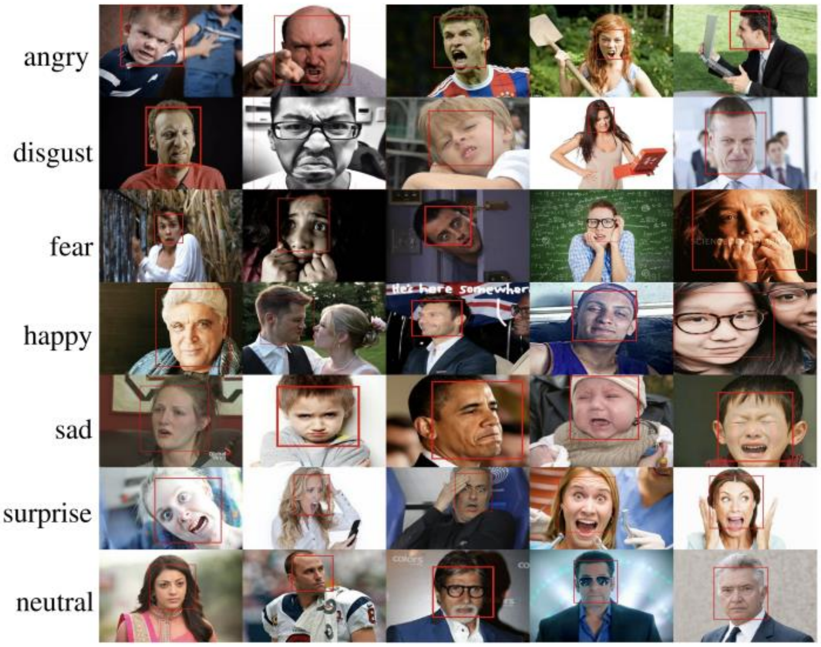


Figure 1: Sample Images from ExpW Dataset

The project can be divided into several stages:

Preprocessing: Apply any necessary processing techniques to remove noise or increase dataset via augmentation, etc. if required.

Training: Split the data into Training/Validation/Test. Use a deep learning framework such as Keras or TensorFlow or any other to train a Convolutional Neural Network on the annotated dataset. You can either create a customized model or employ a pre-trained network (either fine-tuned or as feature extractor). Apply and compare multiple models (3-4) with different hyperparameter settings and do comparative analysis on the performance. For hyperparameter tuning, explore https://keras.io/keras_tuner/.

Testing: The model should be tested on a separate test set of images that was not used for training, to ensure that it can generalize to new data.

Evaluation: Evaluate the performance of the model(s) by measuring its accuracy, precision, recall, and other relevant metrics. Make sure to plot a confusion matrix to determine which expression class is predicted correctly and which is challenging for your model(s).

Requirements:

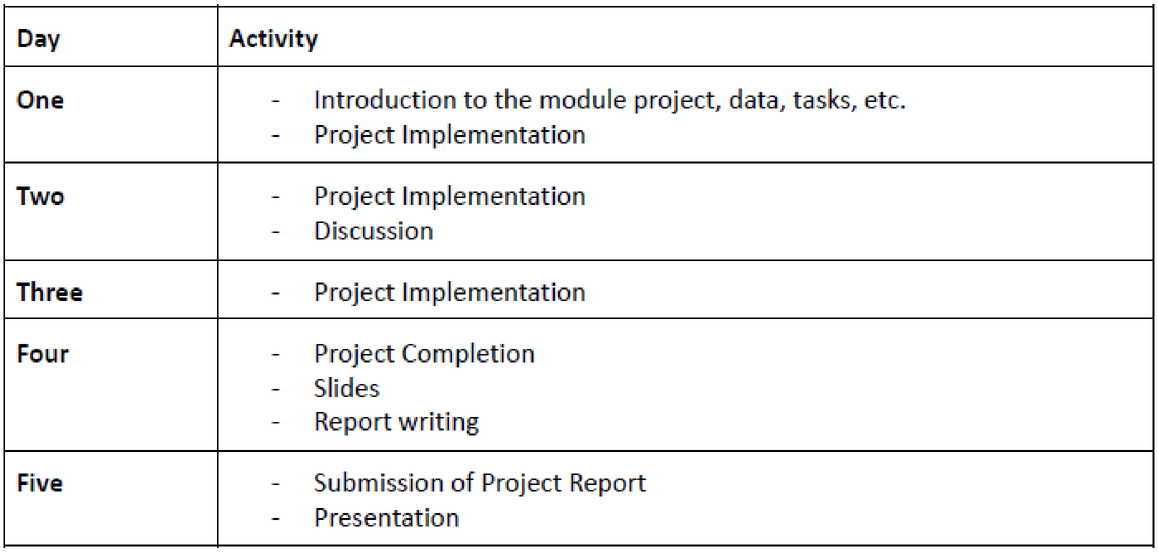
1. Create account on GitHub
2. Share your profile link in the project report (to be submitted at the end of your project)
3. Upload your code to GitHub
4. Prepare a short 3-4 page report about your project that should contain:
    a. Project Title
    b. Your name, email address, github profile link

c. A 100 word summary of your project

d. Project Details
- Overview of the problem and potential application areas
- A brief literature review (refer at least 2 articles from 2022-24) - highlight the work, data, accuracy reported, pros and cons.
- Models you used (architecture, diagram, main components, parameters)
- Dataset you used in the project -stats, data division (training, validation, test)
- Hyperparameter tuning
- Results and Evaluations
- Analysis of results-What are good results? What are bad results and why? A confusion matrix can help provide deeper insight.
- How can you further improve the results?

5. Prepare a 5 slides presentation of your project and the work you have done.
6. Upload your project report and slides to the respective folders in LMS. Alternatively, viva can also be taken by the trainers.

Timeline for Module Project:



Day	Activity
One	- Introduction to the module project, data, tasks, etc. - Project Implementation
Two	- Project Implementation - Discussion
Three	- Project Implementation
Four	- Project Completion - Slides - Report writing
Five	- Submission of Project Report - Presentation

Optional Project

Project Title: Image Data Extraction from Real-Time Traffic Accident Videos

Project Description:

This project involves developing a pipeline to extract and process image data from real-time traffic accident videos for accident detection and analysis. Students will work on transforming live or recorded video streams into labelled image datasets, which can then be used for training machine learning models to recognize and analyse traffic accident events. The project emphasizes preprocessing techniques, frame extraction, and data organization for efficient training and analysis. The extracted image dataset can then be used for further tasks like accident detection, classification, and severity analysis.

Project Objectives:

- **Extract Frames from Video**: Convert traffic accident video streams into sequences of image frames.
- **Pre-process Frames**: Enhance the quality of extracted frames using digital image processing techniques, such as noise reduction and edge detection, to aid in accident detection.
- **Annotate Data**: Manually label frames to identify accident events and provide accurate ground truth for model training.

Key Tasks:

1. **Frame Extraction**: Use tools like OpenCV to capture frames from accident video footage at defined intervals (e.g., 1 frame per second).
2. **Image Pre-processing**: Apply digital image processing methods, such as:
    - **Noise Reduction**: Use Gaussian or median filtering.
    - **Edge Detection**: Apply Canny or Sobel filters to emphasize vehicle structure and possible collision markers.
3. **Data Annotation**: Use tools like LabelImg or VGG Image Annotator (VIA) to label accident events in frames for training a machine learning model.

Suggested Dataset:

- **CADP (Car Accident Detection and Prediction)**: This dataset is highly relevant as it contains real-world video footage of traffic accidents. It includes annotations for accident events, making it ideal for detection and severity classification. Other options include **DAD (Dashcam Accident Dataset)** or **A3D** (Automobile Accident Analysis Dataset) if CADP is unavailable.
- Check these links:
    - GitHub - ankitshah009/CarCrash_forecasting_and_detection
    - DAD Dataset | Papers With Code
    - https://github.com/smallcorgi/Anticipating-Accidents
    - https://github.com/Cogito2012/CarCrashDataset

Additional Tips for Students:

- **Data Augmentation**: Use augmentations such as rotation, brightness adjustment, and flipping to improve model robustness.

- **Cross-Validation**: Apply cross-validation to ensure consistent model performance across different scenes and conditions.
- **Explainability**: Consider using Grad-CAM to visualize the areas of frames contributing most to accident severity classification, adding interpretability to the model’s decisions.

Expected Deliverables:

- **Frame Extraction Script**: Code for transforming video footage into sequential frames.
- **Pre-processed Image Dataset**: Processed images ready for use in accident detection models.
- **Annotated Image Dataset**: Labelled data identifying accident events for machine learning applications.

This project will provide students with hands-on experience in extracting and preparing image data from video footage, a crucial skill in computer vision applications involving real-time data.

Submission is not required!