Data & AI 5/Artificial Intelligence: Deep Learning Project

Alexander Michielsen, Jan Van Sas, Rony Baekeland

Overview

In this project, you will work in groups of three to design, implement, and evaluate a Convolutional Neural Network (CNN) model for classifying emotions from facial images. This project allows you to apply deep learning concepts and showcase your skills in model development, data processing, and analysis.

Objectives

- Design and implement a CNN model to classify emotions from facial images.
- Preprocess and augment image data to improve model performance.
- Train and optimize the model, adjusting hyperparameters as necessary.
- Evaluate the model's performance using appropriate metrics and analysis techniques.
- Document the process and findings in a comprehensive report.

Project Requirements

Note: These are the minimum requirements for a passing grade. You are encouraged to go beyond these requirements to demonstrate your understanding and creativity which will be reflected in your grade. Have a look at the section on possible extensions for inspiration. Please also take into account that part of your final grade will be based on the oral examination of the project.

1. Data Preprocessing and Augmentation

- Obtain and clean a suitable dataset. Multiple datasets exist for this task. A good choice is the FER2013+ dataset. This is an improvement on the FER2013 Dataset by the Microsoft team. It's up to you to explore this dataset and check how FER2013+ is an improvement.
- Apply preprocessing steps such as normalization, and augmentation.

2. Model Development

- Design and implement a CNN architecture tailored for emotion classification.
- Ensure the architecture is modular and easy to debug.

3. Model Training and Optimization

- Train the model using the preprocessed data.
- Tune hyperparameters and apply regularization techniques.
- Log training progress and save checkpoints. Use Tensorboard for this.

4. Evaluation and Analysis

- Assess the model's performance using appropriate metrics.
- Analyze the results to gain insights and provide recommendations for future work.

5. Reporting

- Document the methodology, findings, and challenges together with the code in a clear, well-structured Jupyter notebook.

Deliverables

Interactive Python Notebooks

- Combine code, explanations, and results in well-documented Jupyter Notebooks. All cells should be executed and free of errors.
- Include the following sections within the notebooks:
  - Data Preprocessing: All steps taken to achieve a cleaned and augmented dataset ready for model training.
  - CNN Model Implementation: Source code, including scripts for training and evaluation.
  - Training Logs and Results: Documentation of the training process, including challenges faced and solutions implemented.
  - Analysis: Comprehensive evaluation of the model's performance and insights gained. Include recommendations for future work.

Additional Notes

- Each team member needs to have a complete understanding of all the work that has been delivered.
- You are free to draw inspiration from existing CNN architectures, however, you cannot use a pre-trained model (for the minimum requirements, you can use it when doing the transfer learning). Also, make sure you understand every part of your model.
- You can go as far as you want with this project, however, keep your time management in check. Prioritize if necessary.
- Consider using cloud-based platforms like Google Colab for GPU support.

Possible Extensions

Only look into these once you have completed all aforementioned steps! These are bonus steps for additional marks! If you have any other ideas, feel free to implement them as well.

- **Data**
  - Combine datasets for improved performance.
  - Use face detectors like OpenCV's Haar Cascades or MTCNN.
  - Align and crop images for consistency.
  - **Warning: Advanced, only try if you have time!**: Employ Generative Adversarial Networks (GANs) to create synthetic facial images, expanding the dataset.
- **Model Enhancements**
  - Transfer Learning: Leverage models pre-trained on large datasets (e.g., VGGFace, ResNet) and fine-tune them for emotion classification.

  - Ensemble Methods: Combine predictions from multiple CNN models to create a robust ensemble model

- **Deployment**
  - Deploy the model in a real-time application for live emotion tracking

- **Temporal Dynamics**
  Warning: This is a more advanced extension. Only try if you have time!
  - Extend the project to classify emotions from video clips rather than static images by combining CNNs with RNNs (e.g., LSTM or GRU layers) to capture temporal dependencies

Submission Guidelines

- **Deadline**: *30 december 2024, 23:59*.
- **Submission Method**: Upload your Jupyter Notebooks to Canvas. Ensure that all necessary files are included and that the notebooks can be run without errors.
- **Format**:
  - Notebooks should be named clearly (e.g., `GroupX_DLProject.ipynb`).
  - All code cells should be executed in order, and outputs should be visible.

Good Luck!