Feature Extraction from Images: Approach Outline

Approach:

Data Preprocessing:

- Download images using the provided utilities.
- Normalize and preprocess images (resizing, grayscale conversion).
- Apply data augmentation techniques for better model generalization.

Text Detection & Extraction:

- Use OCR (Tesseract/Google Vision API) to detect text in images.
- Preprocess images for OCR (thresholding, contour detection).
- Extract and parse numeric values and units from the text.

Model Selection:

- Fine-tune a CNN/ResNet model pre-trained on ImageNet to classify entity types (e.g., weight, volume).
- Use regular expressions or rule-based methods to extract and map text to valid units.

Prediction Formatting:

- Ensure extracted values are formatted correctly (e.g., "34 gram").
- Validate predictions against allowed units in constants.py.

Model Evaluation & Optimization:

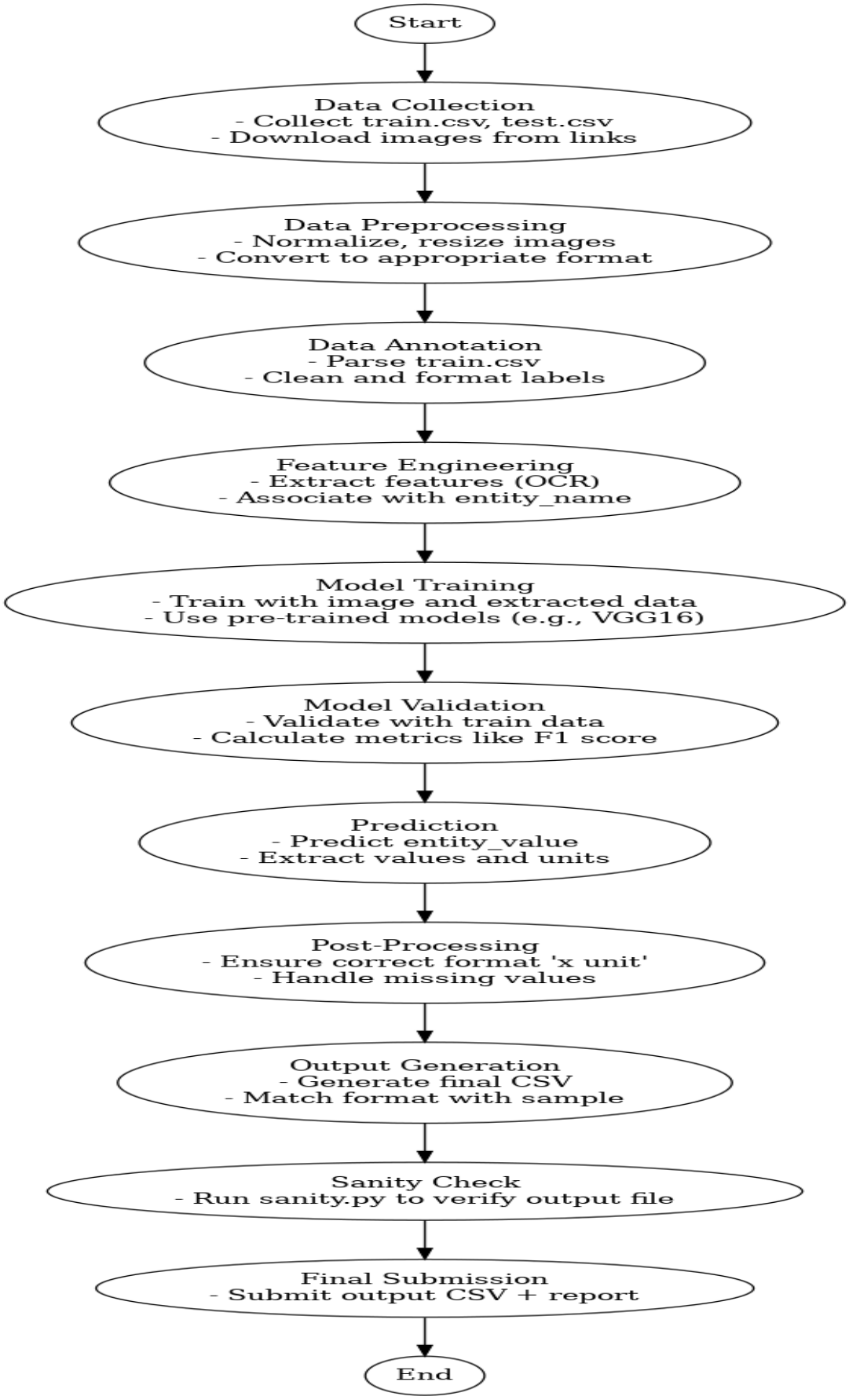
- Evaluate using the F1 score (precision and recall).
- Post-process outputs to ensure correct formatting and unit compliance.

Testing & Sanity Check:

- Use the sanity.py script to ensure predictions match the required output format.
- Generate final predictions for the test dataset, ensuring consistency with sample_test_out.csv.

Tools & Libraries:

- **OCR:** Tesseract, Google Vision API
- **Deep Learning:** PyTorch (for entity recognition)
- **Preprocessing:** OpenCV
- **Data Handling:** pandas, numpy


Start
Data Collection
- Collect train.csv, test.csv
- Download images from links
Data Preprocessing
- Normalize, resize images
- Convert to appropriate format
Data Annotation
- Parse train.csv
- Clean and format labels
Feature Engineering
- Extract features (OCR)
- Associate with entity_name
Model Training
- Train with image and extracted data
- Use pre-trained models (e.g., VGG16)
Model Validation
- Validate with train data
- Calculate metrics like F1 score
Prediction
- Predict entity_value
- Extract values and units
Post-Processing
- Ensure correct format 'x unit'
- Handle missing values
Output Generation
- Generate final CSV
- Match format with sample
Sanity Check
- Run sanity.py to verify output file
Final Submission
- Submit output CSV + report
End