Group ID:24\_25J\_195

Project Title:

1. IT21380914: Developing a Bias Detection Metric For Gender Bias Detection In Audio Datasets.
2. IT21387562: Developing Metric for Measure Gender Bias in Image Datasets Using Contextual Factors: Objects, Scenes, & Spatial Relationships.
3. IT21183690:Developing a Metric for Measure Gender Bias in Contextual Word embeddings.
4. IT21208294: Developing a Metric for Detecting Gender Bias in Human Action Recognition Datasets Through Multi-Dimensional Metrics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | IT21380914 | IT21387562 | IT 21183690 | IT21208294 |
| **.Data Pipeline** | **Data Sources:** | * Mozilla Common Voice Dataset * LibriSpeech Dataset * LibriSpeech Multilingual Dataset * TED-LIUM Dataset * AMI Meeting Corpus | COCO Dataset (filtered for person + gender-associated objects like handbag, skateboard, etc.) | WinoBias Dataset(gender-stereotyped occupation-pronoun sentences) | UCF101 Dataset (Started with “Diving” action category) |
| **Data Preprocessing:** | * Extracted raw audio features: standard deviation of pitch, amplitude, and energy per gender * Computed voice activity duration per gender * Counted number of audio samples for male and female * Stored features in CSV format * Performed controlled data augmentation to simulate synthetic bias * Generated synthetic bias scores based on feature perturbation | * Filtered images with person + selected object categories * Applied YOLOv8 for instance segmentation * Cropped objects and persons; calculated relative size and 3D distance * Generated grayscale depth maps using Depth Anything * Classified gender of the most prominent person * Extracted scene features via CLIP; added Places365 scene labels * Normalized and encoded all features for model input | * Converted token lists to proper sentences. * Tokenized sentences and extracted sentence-level embeddings using BERT. * Computed Cosine Similarity between pronouns and occupations. * Calculated Pointwise Mutual Information (PMI) for gendered word co-occurrences. * Measured Euclidean distances in contextual embeddings. * Normalized all feature values. | * Extracted frames from videos and isolated backgrounds. * Computed CLIP embeddings for scenes. * Classified scenes using Places365. * Computed optical flow for motion vectors * Calculated average speed and motion complexity. * Normalized motion metrics by video duration. |
| **Data storage and versioning** | * Google Drive was used for storing raw audio samples, extracted features, augmented datasets, and model outputs. * Separate folders were maintained for each stage: feature extraction, data augmentation, regression model outputs, and validation results. * CSV files were versioned according to augmentation level (e.g., bias\_5%, bias\_10%, … bias\_50%) for reproducibility and evaluation. | * Google Drive used for storing images, metadata, and model outputs * Separate folders maintained for each step: segmentation, depth, embeddings, scores * JSON and CSV files versioned per approach (UBM v1–v4) for reproducibility | * Used Google Drive to store raw data, preprocessed sentences, embeddings, and results. * Versioned files (JSON, CSV) for feature extraction and bias metric results. | * Google Drive used for storing video frames, motion features, embeddings, and clustering results. * Versioned CSVs maintained for each phase: background analysis, motion analysis, and combined bias scoring. |
| **Model Development** | **Model Selection** | * Symbolic Regression for equation generation attempt one * Polynomial Regression with Ridge (L2) Regularization for equation generation attempt two : the successful equation generation * it worth mentioning for testing purposes in order to derive the WER the model whisper-AI was used | * YOLOv8 for object segmentation * Depth Anything V2 for monocular depth estimation * MobileNet for gender classification * Random Forest Classifier for gender prediction using contextual features * SHAP for feature importance and explainability | * BERT (Pretrained + fine-tuned) for contextual embeddings. * Random Forest for analyzing feature importance * SHAP for explainability and feature attribution. | * CLIP for scene embeddings. Places365 for scene classification. * Random Forest Classifier for gender prediction. * SHAP for comparing feature importance. |
| **Model Training** | * Symbolic Regression trained with evolutionary algorithm (population size = 500, generations = 15) * Polynomial Regression trained using ElasticNet with best alpha = 0.01 and l1\_ratio = 0.0 (pure Ridge) * Validation of regression assumptions (linearity, multicollinearity, etc.) * Final bias score equation selected from Polynomial Regression due to better performance | * Trained gender classification model using labeled image data * Applied SMOTE to balance gender distribution in Random Forest training * Computed SHAP values post-training for interpretability * Trained ML model on features like object size, depth, and scene context | * Fine-tuned BERT to learn gender associations from biased contexts. Evaluated masked token predictions (e.g., whether BERT predicts 'he' or 'she' based on context) * Applied PCA to reduce dimensionality of bias features. * Trained Random Forest Classifier to predict gender bias based on extracted features. * Used SHAP values to validate feature contribution toward bias detection | * Trained gender classification model using video features. * clustering used for scene grouping. |
| **Model Integration** | **Tools used** | * Google Colab * Python * NumPy, Pandas * Matplotlib (for plots and visualizations) * Scikit-learn (for regression models) * Whisper-tiny by OpenAI (for WER validation) * CSV files (for storing datasets, predictions, and results | * Google Colab * Python (OpenCV, pandas, NumPy, scikit-learn, matplotlib, seaborn) * YOLOv8 (Ultralytics), CLIP, Depth Anything V2, Places365, SHAP * Excel/CSV for result aggregation and reporting | * Google Colab * Python (OpenCV, pandas, NumPy, scikit-learn, matplotlib, seaborn) * HuggingFace Transformers (BERT) * SHAP * Excel/CSV for result aggregation and reporting | * Google Colab * Python (OpenCV, Scikit-learn, NumPy, Matplotlib, Seaborn) * CLIP * Places365 scene classifier |
| **Model Deployment** | **Testing Environments** | * All experiments conducted locally using Python * Whisper-tiny model used for validating bias via WER on multiple datasets * Performance validated using MSE, MAE, R², NMSE, Pearson, Spearman, and Kendall correlation metrics * Visualization tools used: correlation plots, bias score trend graphs, WER vs. score comparisons | * Google Colab used for all testing and experimentation * Results validated using statistical tests (Mann-Whitney U, Pearson, Spearman) * Visual inspections done on annotated images, depth maps, and score distributions | * Used Google Colab for all experiments. * Created a biased dataset using WinoBias with gendered occupation-pronoun sentences * Fine-tuned BERT on this dataset to capture gender associations * Evaluated unmasking probabilities (e.g., likelihood of predicting 'he' or 'she' in masked contexts) * Compared BERT’s predictions with the calculated bias metric values for validation * Bias metric validated using statistical tests(Mann-Whitney U Test, Pearson and Spearman Correlation) * Visualized results using boxplots, bar charts, and heatmaps. | * All tests done in Google Colab with GPU support. * Visualized motion and scene clusters via boxplots, histograms, and scatter plots. * Used Mann-Whitney U Test to validate gender-based differences. |
| **Deployment Platform** | * No live deployment yet * Currently designed as an offline evaluation tool with planned open-source release as RESAI TOOLKIT library | * No live deployment yet; results stored and shared via Google Drive * Planned future deployment as part of an open-source bias detection toolkit | * No live deployment yet. * Results stored in Google Drive; intended for integration into future bias detection toolkit | * Results stored and visualized from Google Drive. * Planned deployment in a future open-source video bias evaluation toolkit. |
| **Deployment Method** | * Modular pipeline implemented using Python scripts and Jupyter Notebooks in Google Colab. * Pipeline stages: Feature Extraction → Bias Score Generation * Bias Score will be displayed as the output | * Modular pipeline in Google Colab notebooks * Outputs exported as CSV/XLSX and analyzed through visualizations * Pipeline stages: segmentation → depth → gender → scene → bias scoring | * Modular notebook pipeline: preprocessing → embedding → metric calculation → validation. * Results exported as CSV for interpretation and report generation | * End-to-end pipeline in Colab: video → frame extraction → scene/motion feature generation → clustering → bias analysis Exports results as CSV and visual graphs. |
| **Future Enhancements** | **Model Improvement:** | * Build a web-based GUI to make metric accessible to non-technical users * Extend evaluation to larger and more diverse audio corpora * Incorporate multilingual bias testing more robustly * Explore use of real-world annotations for supervised bias labeling * Integrate tool into a broader fairness assessment toolkit for audio-based AI systems | * Deploy toolkit as a web app with GUI for selecting objects and visualizing gender bias * Automate batch-wise processing by enabling parallel processing of 100 images at once during preprocessing for faster execution | * Extend analysis to intersectional bias (e.g., race + gender). * Develop an interactive web interface to explore gender bias by context. * Add support for multi-language datasets and embeddings. | * Expand analysis to more UCF101 action categories. |