Deep Learning with Multimodal Inputs for Enhanced Issue Resolution

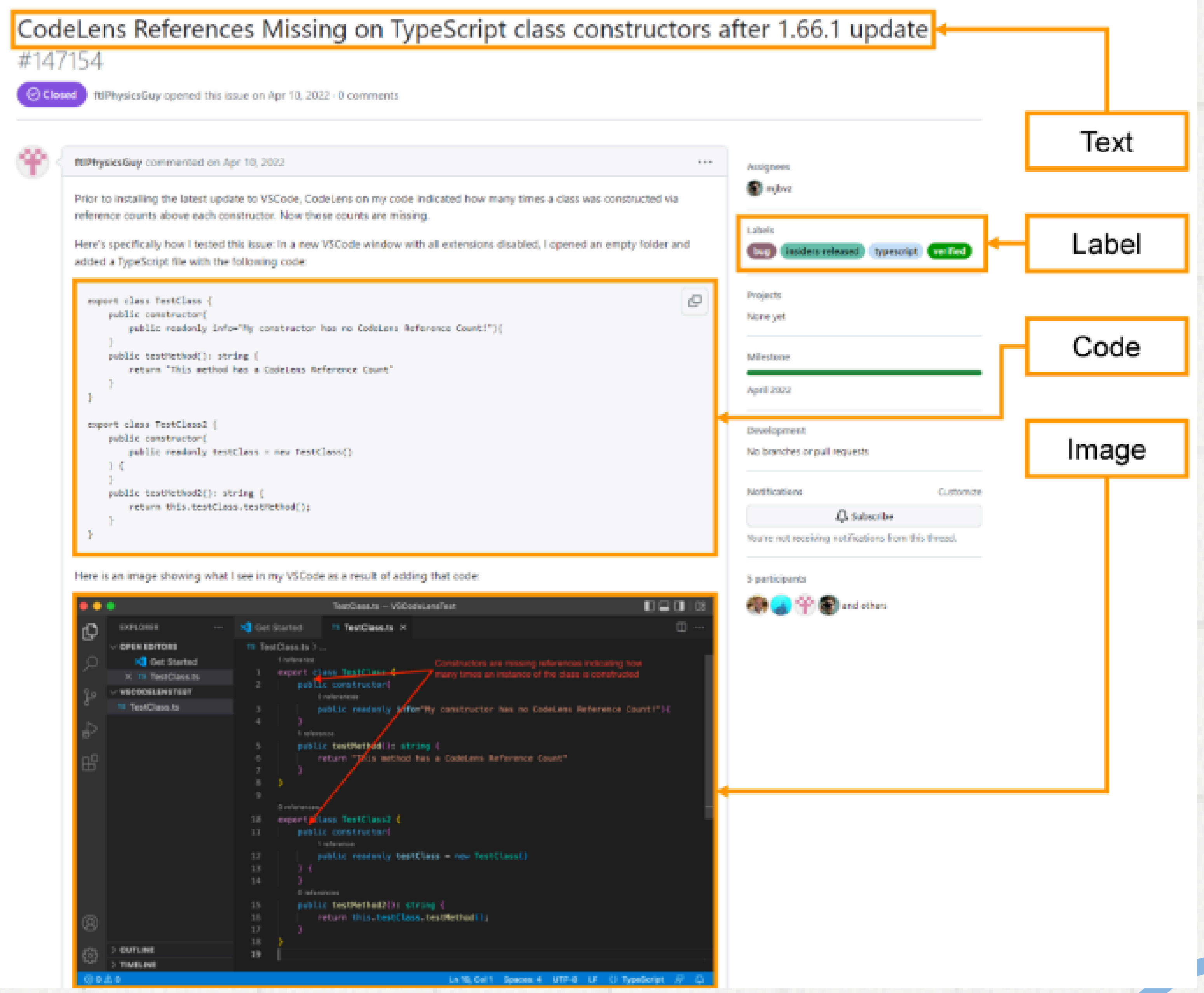
Presented by :
Marina Reda 221101235
Omar adly 221101398
Hazem Ahmed 221100343
Ali sherif 221101562
Loay Gamal 221100419

Field -Program
CSE - AIS

Supervisor
Dr. Mohamed Ghetas

Introduction

- Problem Context:
    - In open-source software development, GitHub's issue tracking system allows developers to report bugs and request features. These reports contain various types of data such as text, code, and images. Traditional classification models primarily use text, missing out on the full context provided by other data types.
- Objective:
    - This project proposes FusionNet, a multimodal deep learning model that combines text, code, and image data to improve the classification of issue reports into bug or feature categories.
- Goal:
    - Evaluate the performance of FusionNet across different data combinations (text, image, code) and demonstrate how leveraging multiple modalities enhances classification accuracy.



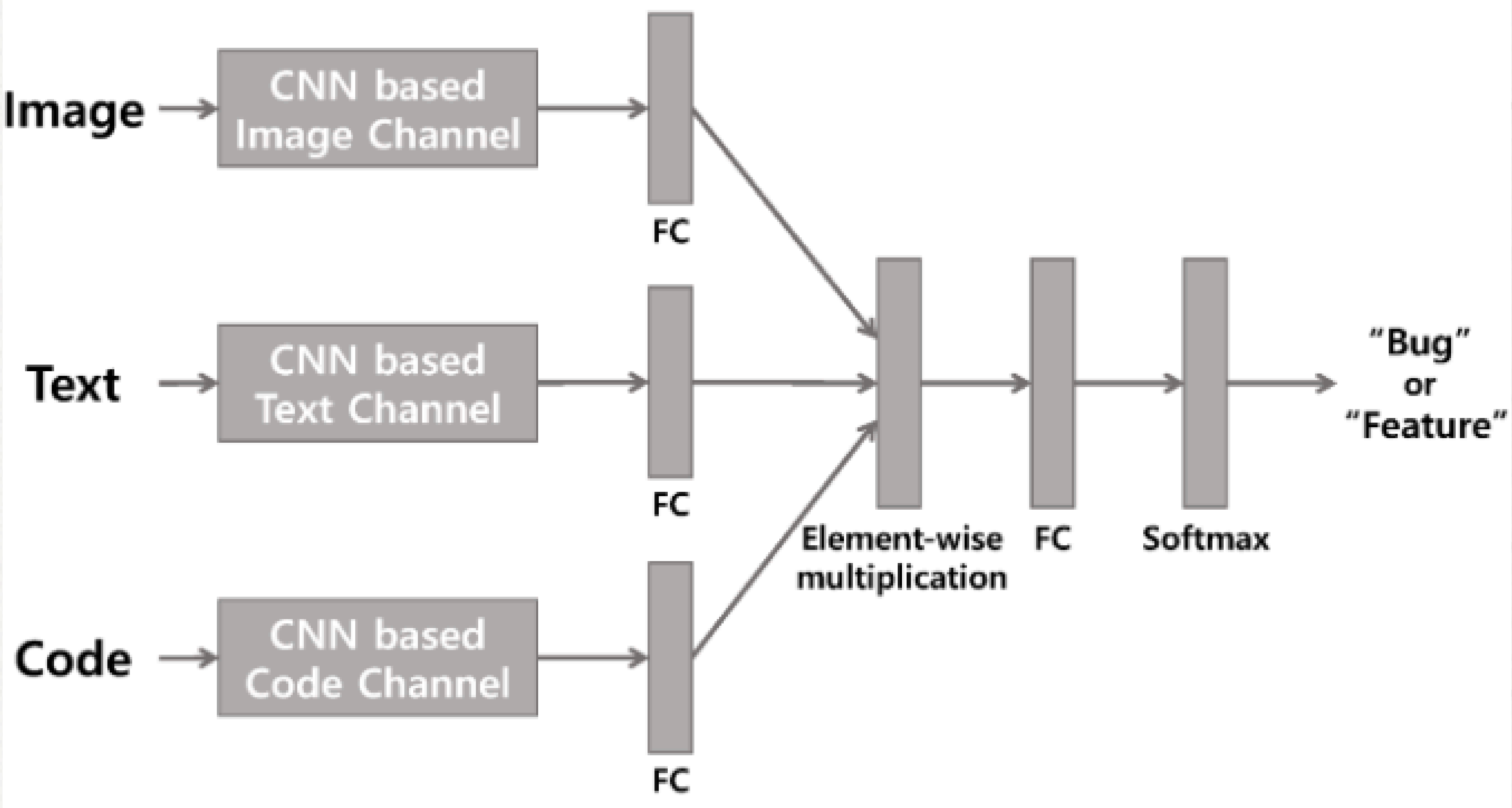
Related Work

- Text-Based Models:
  - Use only text data for classification.
  - Often limited by the inability to capture context from other modalities (images, code).
  - Examples: TF-IDF, RNNs for text classification.
- Multimodal Models:
  - Combine multiple data sources: text, images, and code.
  - Improve classification accuracy by leveraging diverse information.
  - Examples: FusionNet (text + image + code), multimodal CNNs.

Advanced machine learning project _ report.docx

Feature	Text-only Models	Multimodal Models
Data Used	Text only	Text, Image, and Code
Context Understanding	Limited to textual context	Leverages richer context with image and code
Accuracy	Often lower in complex tasks	Improved accuracy, especially in complex tasks
Examples	TF-IDF, RNNs	FusionNet, Multimodal CNNs

The Proposed Multimodal Approach



- FusionNet is a deep learning framework designed for multimodal classification.
- Integrates text, image, and code data for classifying software issue reports.
- Aims to enhance issue report classification accuracy by combining multiple data sources.

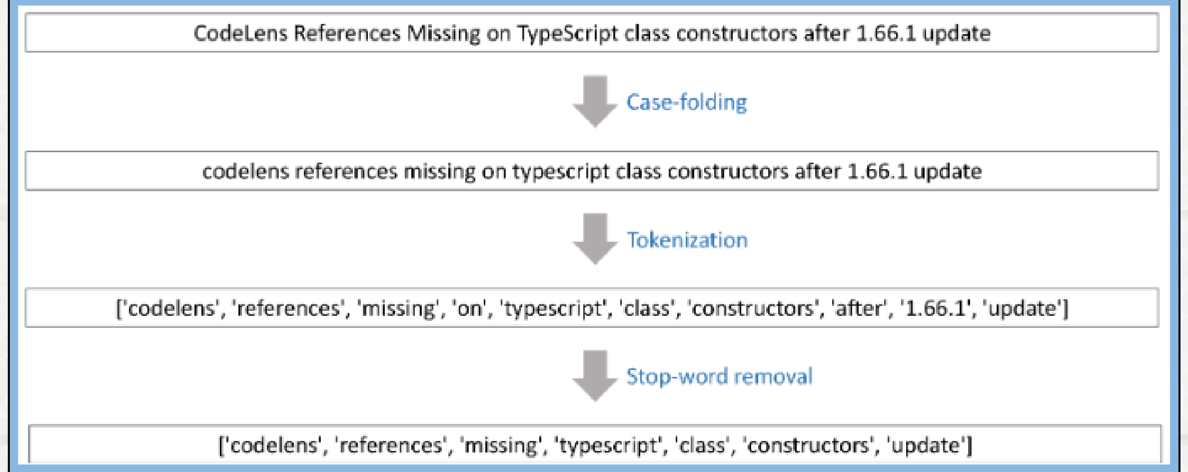
Key Stages of FusionNet:

- Data Pre-Processing: Text, image, and code inputs are pre-processed individually.
- Feature Extraction: Pre-processed data are passed through CNN-based channels to generate feature vectors for each modality.
- Fusion: Feature vectors from all modalities are integrated through element-wise multiplication to create a unified multimodal representation.
- Classification: The fused representation is classified into "bug" or "feature" using a Softmax operation.

Data Pre-Processing

Text Pre-Processing:

- Convert to lowercase (Case-Folding)
- Tokenize text
- Remove stop words and non-alphabetic tokens



Code Pre-Processing:

- Remove comments
- Replace special tokens (\n, \t)
- Convert to lowercase
- Tokenize code

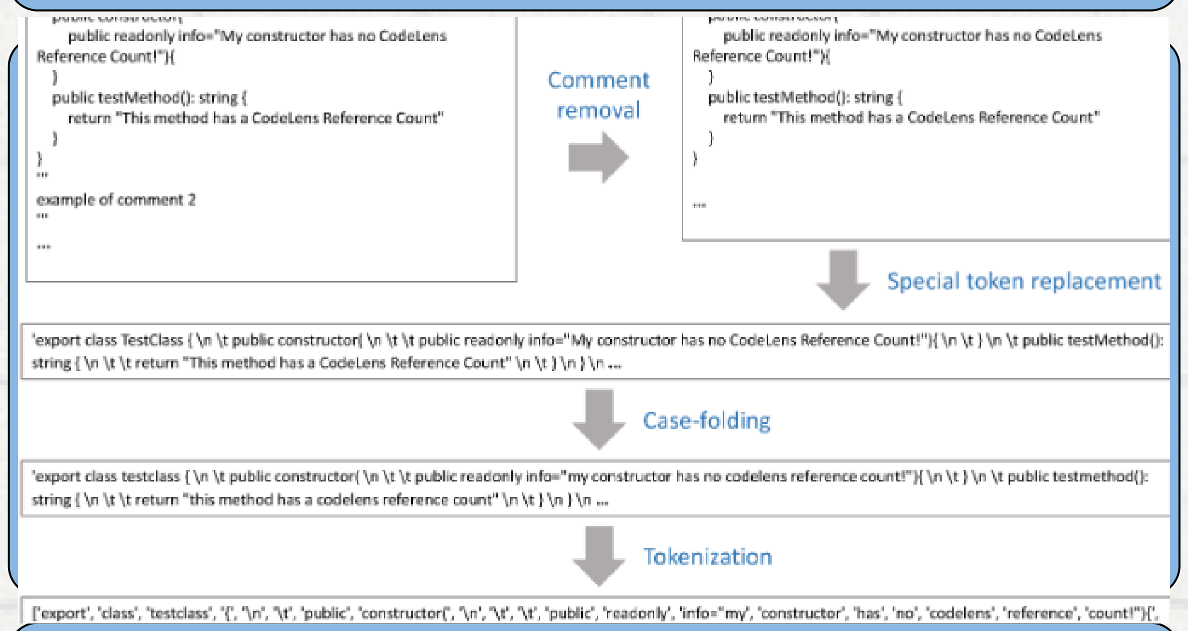
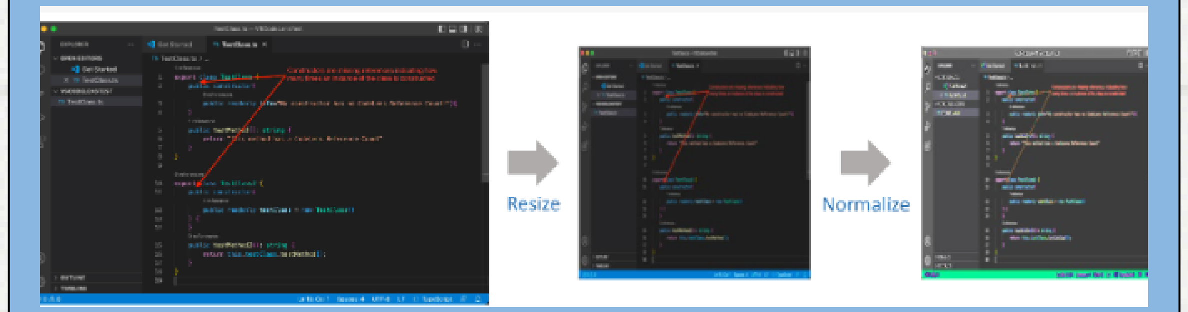


Image Pre-Processing:

- Resize to 258x258 pixels
- Normalize pixel values to [-1, 1]



FEATURE EXTRACTION

Text & Code Data

- Embedding Layer: Converts data into vector representations.
- CNN Processing: Applies convolution, max-pooling, and fully connected layers for feature extraction.

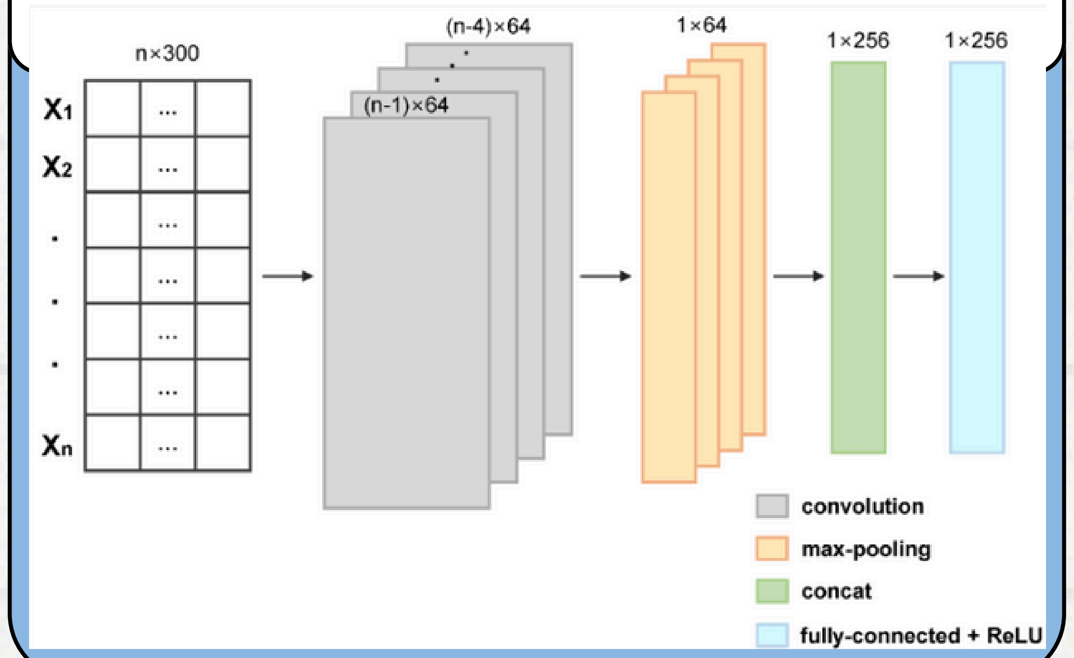
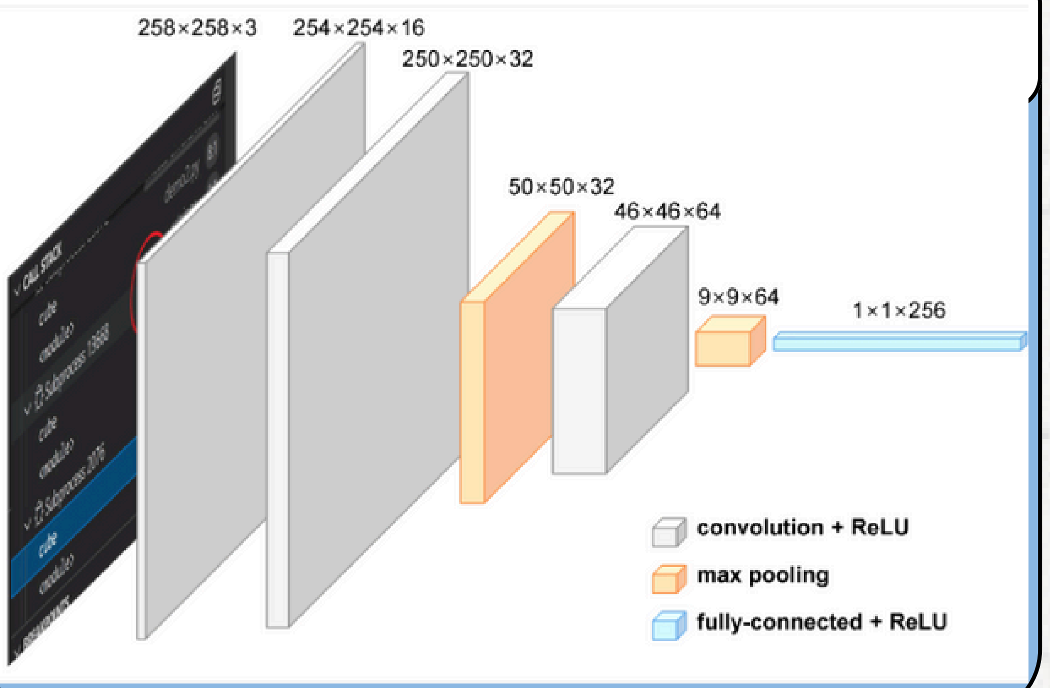


Image Data

CNN Processing: Uses 3 convolution layers, max-pooling, and a fully connected layer to extract features.



KEY STAGES OF FUSIONNET:

Fusion of Feature Vectors

- Fusion Method: Combines features from different modalities (text, code, image) using element-wise multiplication.
- Goal: Integrates data into a unified representation for better learning.
- Benefit: Simplifies and enhances multimodal learning, making classification more accurate.

Classification into "Bug" or "Feature"

- Final Step: Process the fused representation through a fully connected layer.
- Softmax: Produces probabilities for classification into "Bug" or "Feature."
- Optimization: CrossEntropyLoss is used for model optimization.
- Scalability: Can be extended for multiclass classification tasks.

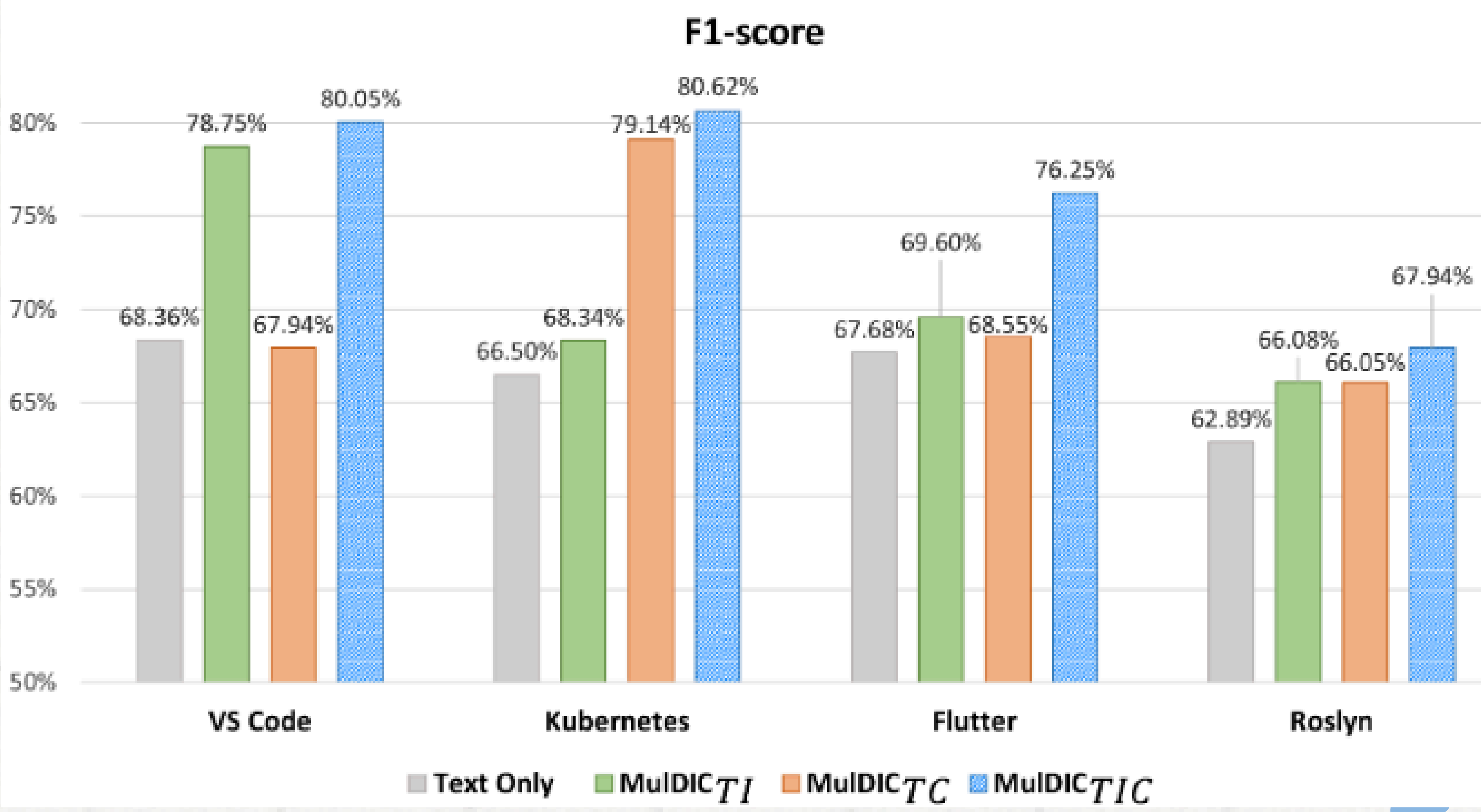
EXPERIMENTAL SETUP

- ***Datasets:***
  - Open-source projects from GitHub (VS Code, Kubernetes, Flutter, Roslyn).
  - Issues labeled as "Bug" or "Feature" for binary classification.
  - Data Sampling: Downsampling applied to balance class distribution (80:20 training/test split).
- ***Models Compared:***
  - Text Only Model: Baseline using only text data.
  - Multimodal Models:
    - FusionNetTI: Text and Image data.
    - FusionNetTC: Text and Code data.
    - FusionNetTIC: Text, Image, and Code data.
- ***Evaluation Metrics:***
  - Precision, Recall, F1-score calculated for each class and averaged based on class distribution.

Project	Total Number of Issues	Label	Num
VS Code	160,218	Bug	28,353
		Feature	20,074
		Total	48,427
Kubernetes	115,035	Bug	13,059
		Feature	5184
		Total	18,243
Flutter	118,576	Bug	13,037
		Feature	9967
		Total	23,004
Roslyn	66,464	Bug	12,882
		Feature	3824
		Total	16,706

RESULT

- Overall Findings:
- FusionNetTIC (Text + Image + Code) outperformed other models across all evaluation metrics (Precision, Recall, F1-score).
- The Text Only model was consistently outperformed by all multimodal models.
- Key Experiments:
- Text + Image (FusionNetTI):
  - Improved Precision (up to 11.15%), Recall (up to 9.65%), and F1-score (up to 10.39%) compared to the Text Only model.
  - Significant improvements seen in VS Code and Roslyn.
- Text + Code (FusionNetTC):
  - Improved Precision (up to 13.06%) and F1-score (up to 12.64%), with VS Code showing slightly lower performance.
- Text + Image + Code (FusionNetTIC):
  - Best performance across all metrics:
    - Precision: +14.34%
    - Recall: +13.90%
    - F1-score: +14.12%



Discussion

01

Multimodal Fusion Benefits:

- The integration of Text, Image, and Code in FusionNet$_{TIC}$ significantly outperformed other models.
- The fusion of diverse data types enriched the classification process, achieving an F1-score improvement of 5.07% to 14.12% over the Text Only model.

02

Text-Image Experiment Insights:

- FusionNet$_{TI}$ (Text + Image) provided valuable improvements, especially in VS Code and Roslyn, showing that images enhance understanding of issue reports.

03

Text-Code Experiment Challenges:

- FusionNet$_{TC}$ (Text + Code) showed mixed results in VS Code, where the Text Only model performed slightly better.
- Potential challenges include:
    - Token Length: No significant data loss was found.
    - Weight Assignment: Equal weights to all modalities may have reduced synergy.
    - Data Quality: Variability in the quality of reports impacted the results.

04

Future Work & Implications:

- Researchers can extend multimodal approaches to multi-class and multi-label tasks.
- Developers and companies can leverage multimodal models to improve issue classification and streamline issue tracking.

Future Directions:

1. State-of-the-Art Feature Extraction: Employ models like BERT for text, CodeBERT for code, and Transformer-based vision models for images to improve accuracy.
2. Advanced Fusion Methods: Explore bilinear pooling techniques like MCB, MLB, MUTAN, and MFB to enhance feature interaction while maintaining computational efficiency.

Conclusions

- FusionNetTIC (Text, Image, Code) outperformed other models, with an F1-score improvement of 5.07% to 14.12%.
- Combining text, image, and code improves issue classification accuracy.
- Multimodal approaches are essential for handling diverse issue report data.

Thank you very much!

AIS TEAM