Visual Document Classification: A Systematic Review of Classical and Modern Methods

Vu Minh Duc, Louisse Tan, Celine Liew,
Tan Xuan Le, Michelle Clara Manik, Akshata J Vedam

Abstract

This project systematically reviews visual document classification methods, covering traditional and modern approaches. We explore image- and text-based models, then advance to multimodal architectures and graph neural networks (GNNs). Experiments with various architectures highlight the strengths and limitations of each approach, emphasizing the need to integrate textual and visual information. We conclude with insights into trade-offs between model complexity, efficiency, and performance.

Image-based Models

Key Features:

- Leverage convolutional layers to extract spatial features and hierarchical patterns.
- **Early architectures**: **AlexNet** (ReLU, dropout, max pooling for feature reduction), **VGG16** (deep 3×3 convolutions for fine-grained textures).
- **Advanced architectures**: **GoogLeNet** (Inception for multi-scale features), **ResNet** (residual connections to enable deep networks).
- Pretrained on **ImageNet** to transfer general visual knowledge.

Text-based Models

Key Features:

- **Bag-of-Words Models**: Logistic Regression, Naive Bayes—use word frequency for text representation.
- **Sequential Models**: LSTM, BiLSTM—capture contextual dependencies using recurrent layers.
- **Transformers**: Self-attention for long-range dependencies; utilize **2D positional encoding** to model spatial relationships.
- **TextCNN**: Applies convolutional operations to extract local patterns and hierarchical text features.

Multimodal Model

Approaches:

- **Stacking**: Combines text and image features using a meta-classifier to improve document classification.
- Uses **VGG16** for image feature extraction and **TextCNN** for text feature extraction, fusing representations at a higher level.
- Provides richer document representations.
- Risk of **overfitting** due to increased model complexity.

Graph Neural Network (GNN)

Key Features:

- Represents document blocks as a graph. Each text box is a node with text feature and image feature.
- **Proximity-based edges** for structural relationships.
- **Feature extraction**: Uses **ResNet18** for images and **averaged GloVe embeddings** for text.
- A **supernode** aggregates global document information.

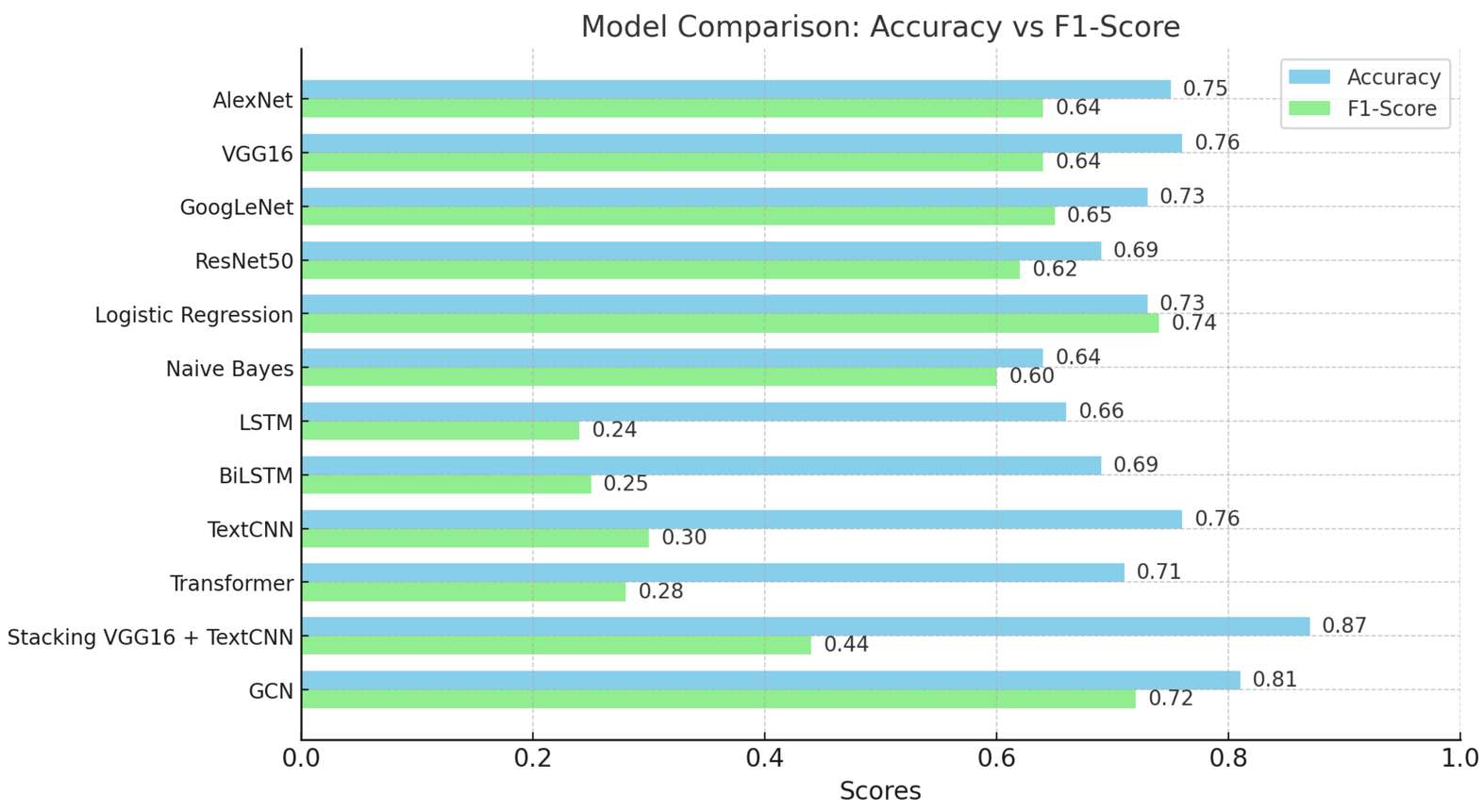
Results and Discussion

Training Setup:

- Dataset: Tobacco-3482, 10 classes.
- Pre-trained weights: ImageNet for images and GloVe embeddings for texts.
- OCR: EasyOCR for text extraction.

Findings and Discussions:

- **Image models** achieve **high accuracy and F1-score**, suggesting that document layout plays a crucial role in classification. However, these models fail to interpret textual content, limiting their effectiveness.
- **Bag-of-words models** provide a **fast, simple, and effective baseline**. Their balance suggests that word frequency-based representations capture sufficient information for classification, but still lack deeper contextual awareness.



- **Sequential models and Transformer** struggle with layout sensitivity, leading to low F1-scores. Concatenating all words disrupts word order, reducing contextual coherence. While 2D positional encoding helps encode spatial relationships, the improvement remains marginal.
- **TextCNN** performs best among single-modal text models, likely because local window capture useful context within individual text boxes. CNNs can extract spatially relevant patterns without losing focus over long sequences.
- **Stacking model** combines strengths from both image-based and text-based models, achieving the **highest accuracy but still suffers from a low F1-score**. This suggests the model may be overconfident in majority-class predictions, leading to poor performance on minority classes.
- **Graph Convolutional Network (GCN)** achieves a **strong balance despite having a much smaller architecture**:
  - **Graph-based representations effectively encode document structure**, capturing both textual and spatial relationships.
  - Supernode aggregation allows the model to generalize better across classes, **reducing overfitting compared to stacked architectures**.