Image Caption Generation using CNN-Based Visual Features and Transformer Language Models

Kabir Man Singh | 24059480

Neural Networks and Deep Learning

Introduction & Problem

- Combining computer vision and NLP to generate meaningful descriptions of images
- Why it matters: Assistive technology, content retrieval, social media automation
- Challenge: Bridging the semantic gap between visual features and natural language
- Goal: Build an end-to-end automatic image caption generation system using deep learning

Dataset & EDA

- Database:Flickr8k
- 8,091 Images | 40,455 Captions | 5 Captions per Image
- Average caption length: 12 words (range 8–15 words)
- Image dimensions: ~400×450px average
- Vocabulary: 5,000 tokens

Dataset & EDA

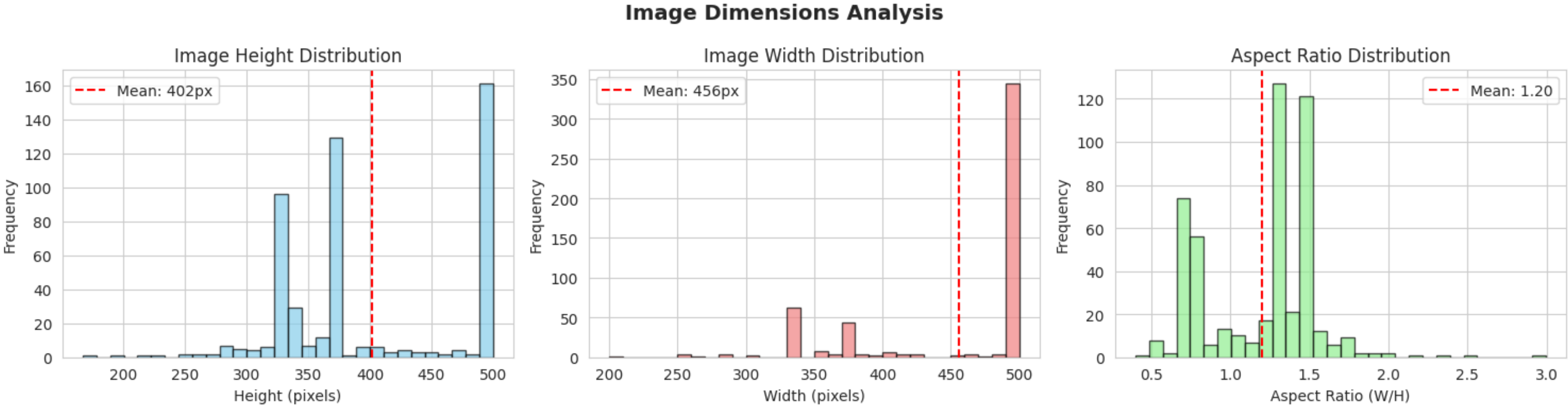
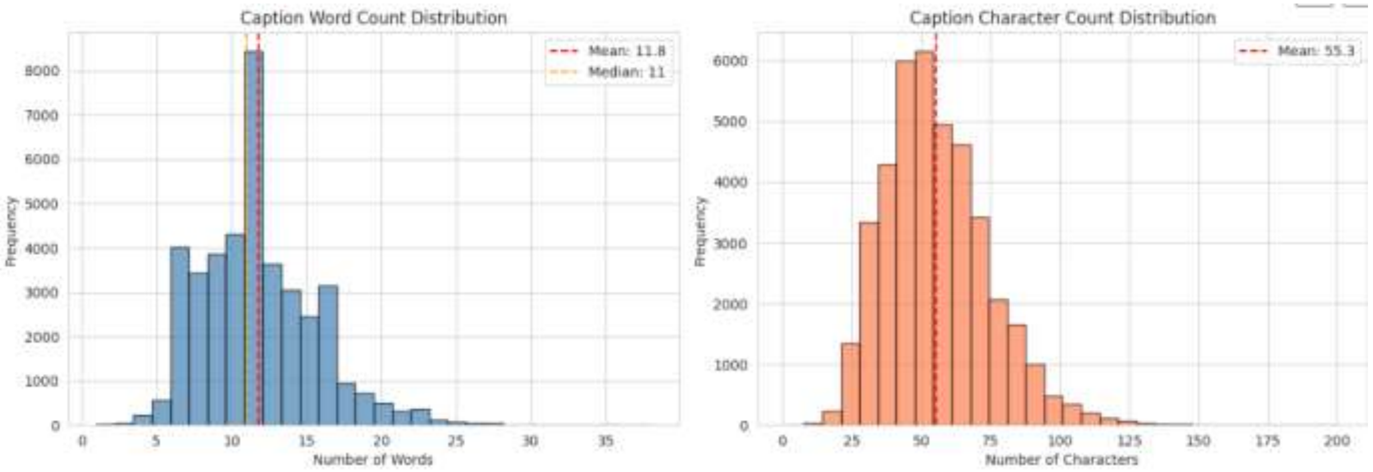
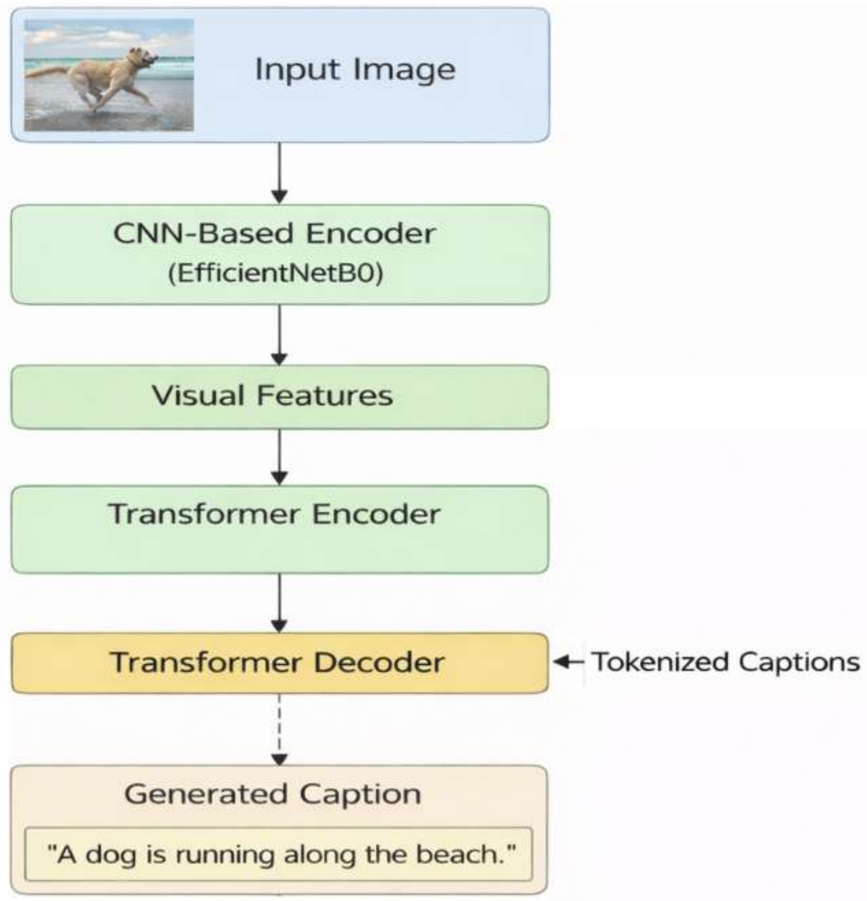


Image Dimensions Analysis
Image Height Distribution
Mean: 402px
Frequency
0
20
40
60
80
100
120
140
160
200
250
300
350
400
450
500
Height (pixels)
Image Width Distribution
Mean: 456px
Frequency
0
50
100
150
200
250
300
350
200
250
300
350
400
450
500
Width (pixels)
Aspect Ratio Distribution
Mean: 1.20
Frequency
0
20
40
60
80
100
120
0.5
1.0
1.5
2.0
2.5
3.0
Aspect Ratio (W/H)


Caption Word Count Distribution
Mean: 11.8
Median: 11
Frequency
0
1000
2000
3000
4000
5000
6000
7000
8000
0
5
10
15
20
25
30
35
Number of Words
Caption Character Count Distribution
Mean: 55.3
Frequency
0
1000
2000
3000
4000
5000
6000
0
25
50
75
100
125
150
175
200
Number of Characters

Methodology & Architecture

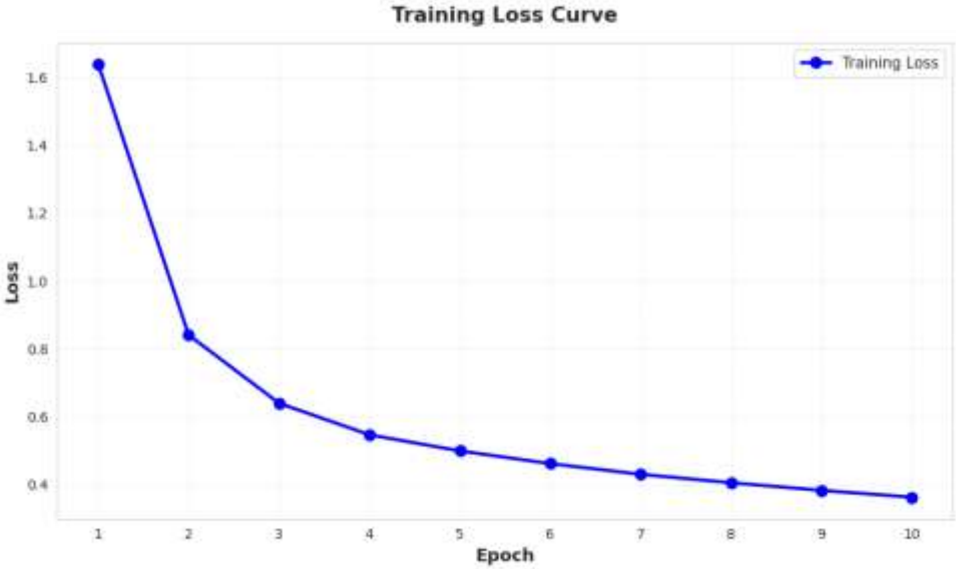
- CNN Encoder: EfficientNetB0 (frozen, pre-trained on ImageNet) → 7×7×1280 feature map → reshaped to 49×1280 → projected to 256D
- Transformer Encoder: 1 layer, 4-head multi-head self-attention + FFN
- Transformer Decoder: Masked self-attention + cross-attention → Dense output layer (5,000 vocab)

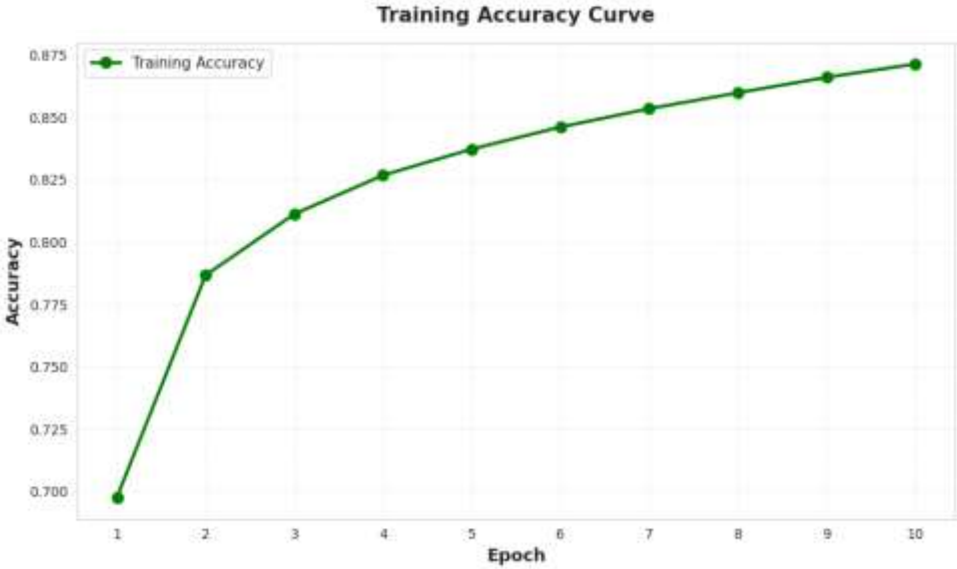


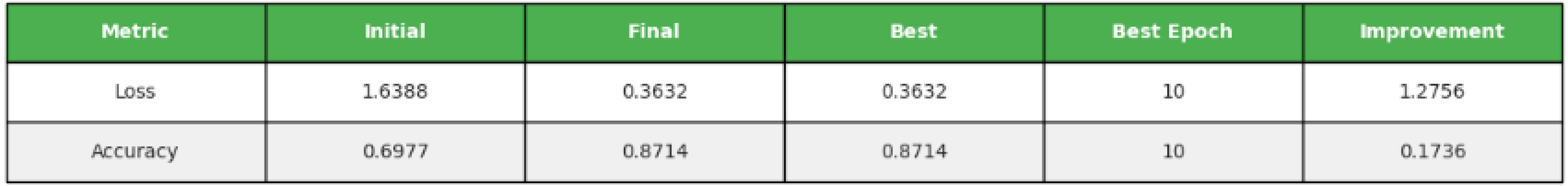
Implementation & Configuration

- Optimizer: Adam (learning rate 0.001)
- Number of Layers: Transformer Encoder (1 layer), Transformer Decoder (1 layer)
- Total Trainable Parameters: 10,495,275
- Total Epochs Trained: 10
- Dataset: Flickr8k (8,091 images, 40,455 captions)
- Loss Function: Sparse Categorical Crossentropy
- Activation Functions: ReLU (FFN), Softmax (output)

Training Results

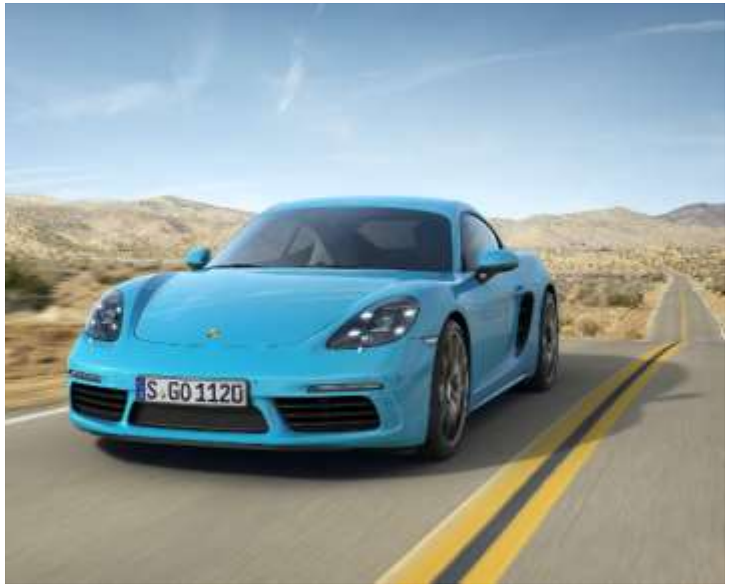

Training Loss Curve

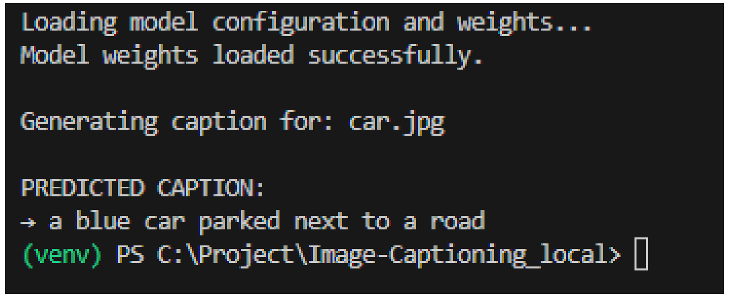

Training Accuracy Curve



Metric	Initial	Final	Best	Best Epoch	Improvement
Loss	1.6388	0.3632	0.3632	10	1.2756
Accuracy	0.6977	0.8714	0.8714	10	0.1736

Visual Evidence

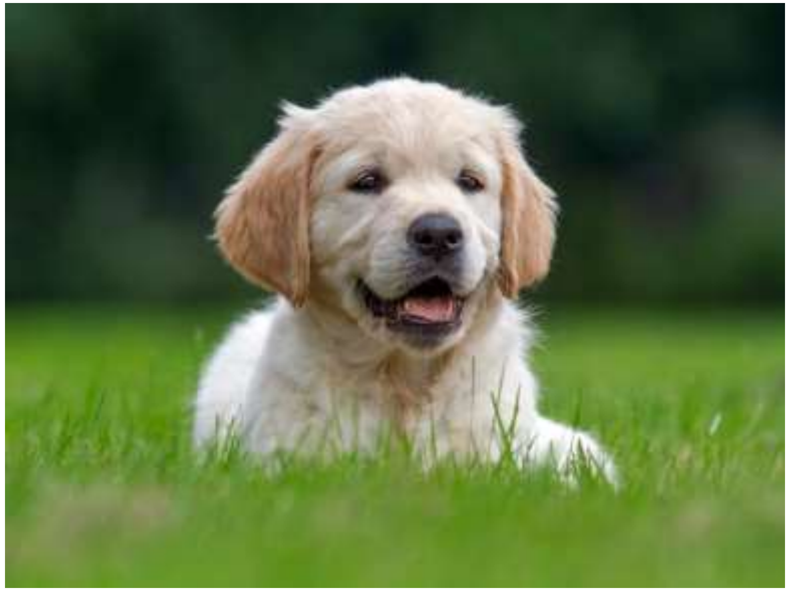


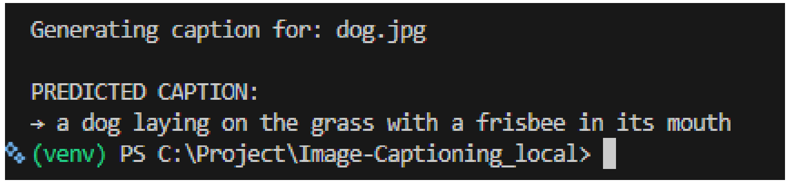


```
Loading model configuration and weights...
Model weights loaded successfully.

Generating caption for: car.jpg

PREDICTED CAPTION:
→ a blue car parked next to a road
(venv) PS C:\Project\Image-Captioning_local>
```





```
Generating caption for: dog.jpg

PREDICTED CAPTION:
→ a dog laying on the grass with a frisbee in its mouth
(venv) PS C:\Project\Image-Captioning_local>
```

Strengths & Weaknesses

Strengths:

- Consistent convergence without overfitting
- Effective transfer learning with EfficientNetB0
- Successful visual-linguistic feature integration
- Resource-efficient (CPU-only training)

Weakness:

- No validation split for generalization testing
- Restricted vocabulary (5,000 tokens)
- No BLEU score evaluation
- Limited performance on complex scenes

Conclusion

- Successfully built end-to-end image captioning system
- Successfully generated captions using deep learning
- Demonstrated practical applications of CV and NLP
- Achieved 87.14% accuracy with 77.8% loss reduction

Future Work

- Use larger datasets (MS COCO)
- Implement train-validation-test splits
- Attention visualization for interpretability

Thank You!