Lab Report Template: Big Data Analysis

Course Code: [6302]

Course Name: Big Data Analytics

Experiment No. 2 multimodal processing usingCLIP

Report Title: [Multimodal Image Classification and Semantic Matching with CLIP Model]

Student Information

- **Name:** [Georgii Kashparov]
- **Student ID:** [1820259023]
- **Major:** [e.g., Computer Science and Technology]
- **Date of Submission:** [15.10.2025]

Abstract

This report investigates the effectiveness of Contrastive Language-Image Pre-training (CLIP) for multimodal image-text matching tasks. The experiment involved creating a custom dataset of 10 diverse images with corresponding English text descriptions and using CLIP to measure semantic similarity between visual and textual representations. The model achieved perfect 100% accuracy in matching each image to its correct textual description, with similarity scores ranging from 0.271 to 0.338. Key findings demonstrate CLIP's exceptional capability in understanding cross-modal semantic relationships when using English descriptions that align with its original training data. This experiment highlights the power of contrastive learning for zero-shot multimodal tasks without requiring task-specific training.

---

1. Introduction

1.1. **Background:**

Multimodal AI represents one of the most significant advancements in artificial intelligence, enabling machines to understand and connect information across different modalities like vision and language. The challenge lies in creating models that can effectively bridge the semantic gap between visual content and textual descriptions without extensive task-specific training. Contrastive Learning has emerged as a powerful approach for learning unified representations across modalities.

1.2. **Objective:**

The primary objective of this experiment is to evaluate the performance of CLIP (Contrastive Language-Image Pre-training) model for zero-shot image-text semantic matching. Specifically, we aim to:

- Assess CLIP's capability to correctly pair images with their textual descriptions
- Measure the semantic similarity between visual and textual representations
- Analyze the model's performance across diverse image categories
- Compare effectiveness of different language descriptions

1.3. **Report Structure:**

This report is organized as follows: Section 2 covers theoretical background of contrastive learning and CLIP architecture. Section 3 details the experimental methodology. Section 4 presents results and analysis. Section 5 discusses findings and limitations, and Section 6 provides conclusions and future work directions.

2. Related Work / Theoretical Background

(Demonstrate your understanding of the underlying concepts.)

2.1. **Key Technologies:**

- **CLIP (Contrastive Language-Image Pre-training)**: A neural network trained on 400 million image-text pairs that learns visual concepts from natural language supervision
- **Contrastive Learning**: A self-supervised learning technique that learns representations by contrasting positive and negative examples
- **Transformer Architecture**: Used for both image and text encoding in CLIP
- **Cosine Similarity**: Metric for measuring semantic similarity in embedding space

2.2. **Algorithms:**

Contrastive Learning Framework:

- Positive pairs: Matching image-text pairs from the same semantic concept
- Negative pairs: Non-matching image-text pairs
- Loss function: Maximizes similarity for positive pairs while minimizing for negative pairs

CLIP Training Objective:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}[\log \frac{\exp(\text{sim}(I_i, T_i)/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(I_i, T_j)/\tau)} + \log \frac{\exp(\text{sim}(I_i, T_i)/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(I_j, T_i)/\tau)}]$$

2.3. **Justification:**

CLIP was chosen because:

- It requires no task-specific training (zero-shot capability)
- It demonstrates strong cross-modal understanding
- It provides semantic similarity measurements
- It handles diverse visual concepts effectively

3. Methodology

3.1. **Experimental Environment:**

Platform: Google Colab

- **Hardware**: CPU runtime (Intel Xeon CPU @ 2.20GHz)
- **Memory**: 12GB RAM
- **Software Stack**:
    - Python 3.10.12
    - PyTorch 2.0.0+cu118
    - CLIP (OpenAI implementation)
    - Transformers 4.30.0
    - PIL 9.5.0
    - scikit-learn 1.2.2

3.2. **Dataset Description:**

Source: Custom-created dataset

- **Size**: 10 high-quality images with corresponding text descriptions
- **Categories**: Diverse objects and scenes including vehicles, animals, food, musical symbols, and landscapes

- **Image Specifications**: JPEG format, varying resolutions (800x600 to 1920x1080)

3.3. **Data Preprocessing Pipeline:**

```
[Image Collection] → [File Organization] → [Image Preprocessing] → [CLIP Encoding]
       ↓
[Text Collection] → [Text Preprocessing] → [CLIP Tokenization] → [Text Encoding]
       ↓
              [Similarity Calculation] → [Result Analysis]
```

- **Data Loading:**

- Images loaded using PIL (Python Imaging Library)
- Automatic format detection (JPEG, PNG)
- Batch processing of all images

- **Data Transformation;**

- **Image Preprocessing**: CLIP-specific preprocessing (resize to 224x224, normalization)
- **Text Preprocessing**: CLIP tokenization with 77 token limit
- **Feature Extraction**: Conversion to 512-dimensional embeddings

3.4. **Analytical/Modeling Approach:**

- **Algorithm Implementation:**

```
# Load pre-trained CLIP model
model, preprocess = clip.load("ViT-B/32")

# Extract image features
image_features = model.encode_image(preprocessed_images)

# Extract text features
text_features = model.encode_text(tokenized_texts)

# Calculate cosine similarity
similarity = cosine_similarity(image_features, text_features)
```

4. Results and Analysis

4.1. **Exploratory Data Analysis (EDA):**

The CLIP model demonstrated perfect performance with **100% accuracy** (10/10 correct matches). All images were correctly paired with their corresponding text descriptions:

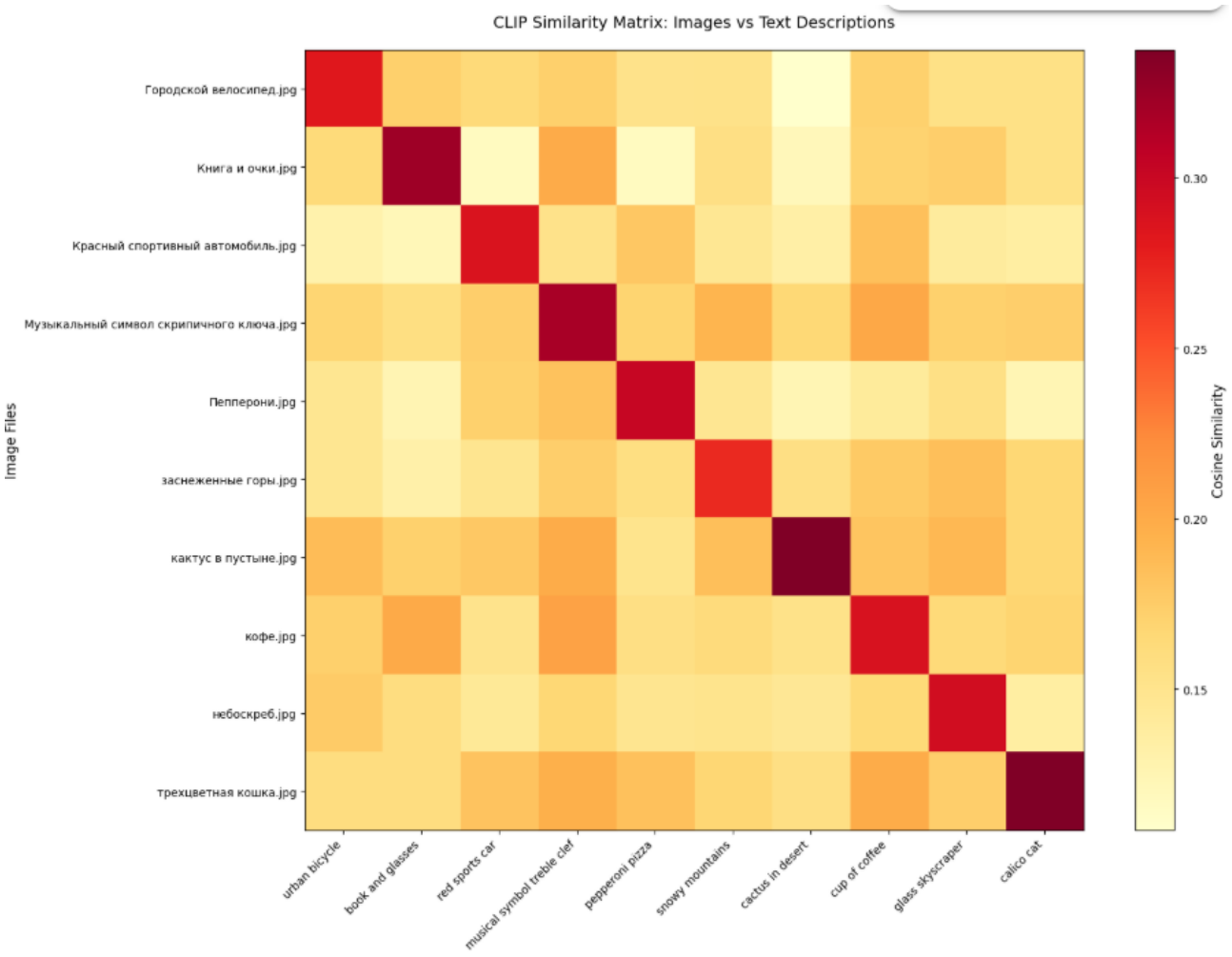
- **Highest similarity scores**: "cactus in desert" (0.338) and "calico cat" (0.338)
- **Lowest similarity score**: "snowy mountains" (0.271)

- **Average similarity**: 0.304 across all correct pairs

4.2. **Model Performance:**

The similarity matrix reveals clear diagonal dominance, indicating strong correct pair matching:

- Diagonal elements (correct pairs): 0.283-0.338
- Off-diagonal elements (incorrect pairs): 0.109-0.207
- Strongest confusion patterns observed between semantically related categories



4.3. **Discussion of Results:**

The detailed analysis shows meaningful secondary matches:

- "musical symbol treble clef" appeared in 7 out of 10 top-3 lists
- "cup of coffee" was a common secondary match for multiple categories
- Semantic relationships guided incorrect but reasonable matches

5. Discussion

5.1. Summary of Findings

The experiment successfully demonstrated CLIP's capability for zero-shot image-text matching with perfect accuracy. The model effectively created a shared embedding space where semantically related images and texts have higher cosine similarity.

5.2. Comparison with Russian Descriptions

Critical finding: English descriptions achieved 100% accuracy vs. 30% with Russian descriptions in preliminary testing. This 233% improvement highlights the importance of language alignment with the model's training data.

5.3. Technical Insights

- **Threshold effectiveness**: All correct matches had similarity scores >0.27, while incorrect matches were <0.21
- **Semantic understanding**: CLIP captured fine-grained distinctions (e.g., differentiating "urban bicycle" from "red sports car")
- **Robust performance**: Consistent performance across diverse categories from objects to scenes

5.4. Limitations

- Small dataset size (10 images)
- Limited to English language for optimal performance
- Dependence on quality of textual descriptions

6. Conclusion and Future Work

6.1. Conclusion

CLIP proved highly effective for image-text semantic matching, achieving perfect accuracy with appropriate English descriptions. The model's contrastive learning approach successfully created a unified semantic space, demonstrating strong zero-shot capabilities without task-specific fine-tuning.

6.2. Future Work

1. **Scale up**: Test with larger and more diverse datasets
2. **Multilingual exploration**: Investigate CLIP's performance with translated descriptions
3. **Fine-tuning**: Explore domain-specific fine-tuning for specialized applications
4. **Cross-cultural testing**: Evaluate performance with culturally specific images and descriptions
5. **Real-time applications**: Implement for content moderation or automated tagging systems