

UNIVERSITY OF FRIBOURG

MASTER THESIS

---

# Contrastive Learning for Character Detection in Ancient Greek Papyri

---

*Author:*  
Vedasri Nakka

*Supervisor:*  
Prof. Rolf Ingold  
Prof. Andreas Fischer  
Lars Vögtlin

*in the*  
Department of Informatics  
Document, Image and Video Analysis(DIVA)

October 10, 2024

## Abstract

This thesis investigates the effectiveness of SimCLR [6], a contrastive learning technique, specifically in the context of Greek letter recognition, and examines the impact of various augmentation techniques. To achieve this, we use a large Alpub dataset [60] (pretraining dataset) to pretrain the SimCLR backbone, followed by fine-tuning on a smaller ICDAR [56] dataset (finetuning dataset) to evaluate the performance of SimCLR in comparison to traditional baseline models using cross-entropy and triplet loss functions. Furthermore, our work explores the impact of several data augmentation strategies, a critical component of the SimCLR training pipeline.

Methodologically, our study examines three primary approaches: **(1)** a Baseline model with cross-entropy loss, **(2)** a Triplet embedding model, enhanced with a classification layer, and **(3)** a SimCLR pretrained model with a classification layer. Initially, we train the baseline model, triplet model, and SimCLR with 93 different augmentations on ResNet-18 and ResNet-50 networks [25] using the ICDAR dataset. From these, we select the *top-4* augmentations based on the results of a statistical t-test. Finally, we conduct pretraining of SimCLR on the large Alpub dataset, followed by fine-tuning on the smaller ICDAR dataset. The triplet loss model undergoes a similar training process, being pretrained on the top-4 augmentations using the Alpub dataset, and then fine-tuned on the ICDAR dataset.

Our experiments reveal that SimCLR does not outperform the baselines in letter recognition tasks. The baseline model using cross-entropy loss demonstrates superior performance compared to both SimCLR and the triplet loss method. This study provides a detailed evaluation of contrastive learning for letter recognition and highlights the limitations of SimCLR, emphasizing the effectiveness of traditional supervised learning models in this specific application. We believe that the cropping strategies involved in SimCLR lead to a semantic shift of the input image, thereby reducing the effectiveness of training, despite the large amount of pretraining data used. Our code is available at [https://github.com/DIVA-DIA/MT\\_augmentation\\_and\\_contrastive\\_learning/](https://github.com/DIVA-DIA/MT_augmentation_and_contrastive_learning/).

**Keywords:** Contrastive Learning, Image Recognition, Greek papyri, SimCLR, Triplet loss, ResNet18, ResNet50, Augmentations, Charater recognition

# Contents

<b>Abstract</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>4</b>
<b>3 Methodology</b>	<b>6</b>
3.1 Primary Augmentations . . . . .	6
3.2 Training Methods . . . . .	11
3.2.1 Baseline Model . . . . .	11
Data Augmentation . . . . .	11
Cross-Entropy Loss . . . . .	12
Overall Training Pipeline . . . . .	13
3.2.2 Triplet Model . . . . .	13
Architecture . . . . .	13
Triplet Loss . . . . .	14
Data Augmentation . . . . .	15
Overall Training Pipeline . . . . .	16
3.2.3 SimCLR . . . . .	17
Data Augmentations . . . . .	18
Overall Training Pipeline . . . . .	18
3.3 Summary . . . . .	21
<b>4 Experiments</b>	<b>22</b>
4.1 Experimental Setting . . . . .	22
4.1.1 Datasets . . . . .	22
4.1.2 Implementation Details . . . . .	22
4.1.3 Methods . . . . .	23
4.1.4 Data Augmentations . . . . .	24
4.2 Results . . . . .	27
4.2.1 Results without Pretraining on Alpub . . . . .	28
4.2.2 Results with Pretraining . . . . .	28
4.3 t-SNE Analysis . . . . .	32
4.4 Summary . . . . .	33
<b>5 Discussion and Limitations</b>	<b>36</b>
5.1 Discussion . . . . .	36
5.2 Limitations . . . . .	37
<b>6 Conclusion &amp; Future Work</b>	<b>39</b>
6.1 Conclusion . . . . .	39
6.2 Future Work . . . . .	39

<b>A Full Results of Experiments</b>	<b>47</b>
A.1 Baseline Model Results . . . . .	47
A.1.1 ResNet-18 Architecture . . . . .	47
A.1.2 ResNet-50 Architecture . . . . .	47
A.1.3 ALPUB Dataset Evaluation . . . . .	47
A.2 Triplet Model Results . . . . .	47
A.2.1 ResNet-18 Architecture . . . . .	47
A.2.2 ResNet-50 Architecture . . . . .	48
A.2.3 ALPUB Dataset Evaluation . . . . .	48
A.3 SimCLR Model Results . . . . .	48
A.3.1 ResNet-18 Architecture . . . . .	48
A.3.2 ResNet-50 Architecture . . . . .	48
A.3.3 ALPUB Dataset Evaluation . . . . .	48
A.4 Baseline Results with Cross-Entropy Loss Using ResNet-18 and ResNet-50 Networks . . . . .	48
A.4.1 Baseline using ResNet-18 architecture on ICDAR Dataset . . . . .	48
A.4.2 Baseline using ResNet-50 architecture on ICDAR Dataset . . . . .	50
A.4.3 Baseline using ResNet-18 architecture pertaining on Alpub Dataset . . . . .	52
A.4.4 Baseline using ResNet-18 architecture pertaining on Alpub Dataset & fine-tuning on ICDAR dataset . . . . .	53
A.4.5 Baseline using ResNet-50 architecture on Alpub Dataset . . . . .	53
A.4.6 Baseline using ResNet-50 architecture on Alpub Dataset & fine-tuning on ICDAR dataset . . . . .	54
A.5 Baseline with Triplet embedding model Using ResNet-18 and ResNet-50 Networks . . . . .	54
A.5.1 Triplet Model using ResNet-18 Pre-training on ICDAR Dataset . . . . .	54
A.5.2 Triplet Model Using ResNet-18 fine-tuning on ICDAR Dataset Without Backbone . . . . .	56
A.5.3 Triplet Model Using ResNet-18 fine-tuning on ICDAR Dataset With Backbone . . . . .	58
A.5.4 Triplet model using ResNet-18 Pre-training on Alpub dataset . . . . .	61
A.5.5 Triplet Model Using ResNet-18 fine-tuning on Alpub Dataset Without Backbone . . . . .	61
A.5.6 Triplet Model Using ResNet-18 fine-tuning on Alpub Dataset With Backbone . . . . .	61
A.5.7 Triplet model using ResNet-50 Pre-training on ICDAR dataset . . . . .	62
A.5.8 Triplet Model Using ResNet-50 pretrain on ICDAR dataset fine-tuning on ICDAR Dataset Without Backbone . . . . .	64
A.5.9 Triplet Model Using ResNet-50 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset With Backbone . . . . .	66
A.5.10 Triplet model using ResNet-50 Pre-training on Alpub dataset . . . . .	68
A.5.11 Triplet Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset Without Backbone . . . . .	69
A.5.12 Triplet Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset With Backbone . . . . .	69
A.6 SimCLR model Using ResNet-18 and ResNet-50 Networks . . . . .	69
A.6.1 SimCLR model using ResNet-18 Pre-training on ICDAR dataset . . . . .	69
A.6.2 SimCLR Model Using ResNet-18 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset Without Backbone . . . . .	72
A.6.3 SimCLR Model Using ResNet-18 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset With Backbone . . . . .	74

A.6.4 SimCLR model using ResNet-18 Pre-training on Alpub dataset	76
A.6.5 SimCLR Model Using ResNet-18 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset Without Backbone . . . . .	76
A.6.6 SimCLR Model Using ResNet-18 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset With Backbone . . . . .	77
A.6.7 SimCLR model using ResNet-50 Pre-training on ICDAR dataset	77
A.6.8 SimCLR Model Using ResNet-50 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset Without Backbone . . . . .	79
A.6.9 SimCLR Model Using ResNet-50 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset With Backbone . . . . .	81
A.6.10 SimCLR model using ResNet-50 Pre-training on Alpub dataset	83
A.6.11 SimCLR Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset Without Backbone . . . . .	84
A.6.12 SimCLR Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset With Backbone . . . . .	84

# List of Abbreviations

<b>SimCLR</b>	Simple Framework for Contrastive Learning of Visual Representations
<b>ResNet-18</b>	Residual Neural Network - 18
<b>ResNet-50</b>	Residual Neural Network - 50
<b>ICDAR</b>	The International Conference on Document Analysis and Recognition
<b>InfoNCE</b>	Noise-Contrastive Estimation
<b>CE</b>	Cross Entropy loss
<b>ALPUB</b>	Ancient Lives Project for University of Basel

## Chapter 1

# Introduction

In recent years, contrastive learning [36, 51, 62, 66, 71] has emerged as a powerful unsupervised learning technique [3, 19, 22, 13, 17, 67] within the field of computer vision, offering promising results across various applications. Among these methods, SimCLR (Simple Framework for Contrastive Learning of Visual Representations), introduced by Chen et al. [6], has shown remarkable success in a range of image recognition tasks by leveraging the power of contrastive pretraining. As a result, SimCLR has been extended to a variety of computer vision tasks, including semantic segmentation [33, 28, 73], object detection [7, 44, 68], and object tracking [38, 35], among others. However, its application in more specialized domains, such as letter recognition, has not been thoroughly explored. This thesis seeks to address this gap by investigating the effectiveness of SimCLR in Greek letter recognition, specifically through pretraining on the large-scale Alpub dataset [60], followed by fine-tuning on the domain-specific ICDAR dataset [56].

**Research Questions.** The central question guiding this research is: *How effective is SimCLR contrastive pretraining for letter recognition tasks compared to traditional baseline methods, and which data augmentations enhance its performance?* Specifically, we seek to determine whether SimCLR can outperform baseline models that are traditionally trained using cross-entropy and triplet loss functions in distinguishing between various classes of letters.

**The primary objectives of this study are:**

- To assess the performance of SimCLR in letter recognition tasks using both a large dataset (ALPUB) and a smaller ICDAR dataset.
- To compare SimCLR’s performance with that of baseline models trained with cross-entropy and triplet loss.
- To identify effective data augmentations that improve the performance of SimCLR and baseline models.

To systematically investigate these objectives, we structure our analysis across the following chapters:

- **Chapter 2:** Related Work reviews existing literature, covering essential topics such as handwritten character recognition, various neural architectures, contrastive learning methods, triplet loss, and data augmentation techniques.
- **Chapter 3:** Methodology outlines the experimental design and the approach taken to address our research questions.



**FIGURE 1.1: Reference images.** Sample full images from the ICDAR dataset containing Greek letters. We use ground-truth annotations to crop each letter subimage from the full images. We train models on the cropped letter images, and finally evaluate the performance on unseen test data.

- **Chapter 4:** Results presents the outcomes of our experiments, focusing on the comparative performance of the different methods.
- **Chapter 5:** Discussion offers a detailed analysis of the results, exploring the implications of our findings.
- **Chapter 6:** Conclusion summarizes the thesis, discusses the limitations encountered, and suggests directions for future research.

**Our Contributions.** This thesis contributes to the field by exploring the application of SimCLR contrastive pretraining in letter recognition, a task that requires precise classification of letter forms. We focus on two datasets: the ALPUB dataset, comprising 24 classes of letters, and a smaller ICDAR dataset, which includes 25 classes with 153 full-size training images and 34 test images, which are cropped for every letter to obtain a total of 34,061 cropped images. An example of Greek papyri images is shown in Figure 1.1. The study employs three distinct methods to assess performance:

1. A baseline model trained with cross-entropy loss
2. A triplet model trained with triplet loss
3. A SimCLR model trained with InfoNCE loss

By comparing these methods, we aim to understand the relative performance of SimCLR against traditional baselines and to explore whether contrastive learning offers significant advantages in letter recognition tasks.

The experimental process begins with training the baseline models using 93 different augmentations on ResNet-18 and ResNet-50 networks [25]. The top-4 augmentations are selected based on their performance, after which the SimCLR model is trained with these augmentations on the large Alpub dataset [60] and fine-tuned on the smaller

ICDAR dataset [56]. The triplet loss model follows a similar procedure, being trained and fine-tuned with the selected augmentations on both datasets, using the ResNet-18 and ResNet-50 architectures.

Our findings indicate that SimCLR does not surpass traditional methods in letter recognition. The baseline model using cross-entropy loss outperforms both SimCLR and the triplet loss model. This study thus provides valuable insights into the limitations of SimCLR for letter recognition tasks and highlights the continued relevance of traditional supervised learning models in this domain. By systematically evaluating these methods, we contribute to the broader research in computer vision and offer practical insights for improving letter recognition systems.

## Chapter 2

# Related Work

In this chapter, we will review the literature related to our work. We have segregated the works based on their themes and identified four distinct directions.

**Letter Recognition.** Handwritten character recognition, particularly for specific alphabets like Greek [47, 61], has been a research focus for many years. Early approaches relied on traditional machine learning techniques with hand-crafted features. Techniques such as Scale-Invariant Feature Transform (SIFT) [46], Histogram of Oriented Gradients (HOG) [10], and Local Binary Patterns (LBP) [49] were commonly used to extract features from images. These methods, however, required significant domain expertise and often struggled to generalize across different handwriting styles and scripts [54, 64].

The advent of deep learning around 2013 revolutionized character recognition [41, 70], leading to substantial improvements in accuracy and robustness. Convolutional Neural Networks (CNNs) [42, 69] have since become the backbone of modern character recognition systems. Advanced architectures such as ResNet [25, 26], VGG [58], and DenseNet [34] have demonstrated exceptional performance in various image recognition tasks, including handwritten character recognition [48, 61]. These models typically use cross-entropy loss for classification [12], which is widely adopted due to its simplicity and effectiveness. Nevertheless, traditional supervised learning methods have their limitations. They require large amounts of labelled data and can be prone to overfitting, especially with small or imbalanced datasets [1]. These challenges have driven researchers to explore alternative approaches that utilize data more efficiently and improve model generalization without heavy reliance on labelled datasets [6, 24, 20, 74, 3].

**Contrastive Learning.** Contrastive learning methods, like SimCLR [6], have emerged as powerful techniques in self-supervised learning. These methods train models to differentiate between similar and dissimilar data pairs. SimCLR, in particular, uses extensive data augmentations to create positive pairs (views of the same image) and negative pairs (views of different images), enabling the learning of feature representations without labeled data. The effectiveness of SimCLR has been demonstrated across various visual recognition tasks [16, 21], often outperforming traditional supervised learning methods. However, its application in handwritten character recognition, particularly in non-Latin scripts like Greek, has not been thoroughly explored. Most research has focused on commonly used datasets such as CIFAR-10 [40] and ImageNet [41], leaving a gap in understanding how contrastive learning methods perform in more specialized contexts [51, 37].

**Triplet Loss.** Triplet loss, introduced by Hoffer and Ailon (2015) [31], is another method used for learning discriminative embeddings by comparing an anchor image with a positive (similar) and a negative (dissimilar) image. This approach has been widely applied in tasks like face recognition [55, 4], person reidentification [72, 29], Image retrieval [11, 14], fine-grained image recognition [15] and has shown promise in character recognition as well. Triplet loss aims to bring embeddings of similar examples closer in the embedding space while pushing dissimilar ones apart. Despite its simplicity and effectiveness, triplet loss methods face challenges such as careful triplet selection and high computational cost during training [30]. These challenges have limited its widespread adoption in character recognition tasks, especially when dealing with large and complex datasets [2].

**Data Augmentation.** Data augmentation is essential for enhancing the generalization capabilities of machine learning models, particularly in visual tasks. Augmentations introduce variability into the training data, enabling models to learn more robust and diverse features. The Albumentations library [5] provides a comprehensive suite of augmentation techniques, including both spatial and pixel-level transformations, which have been effectively applied to tasks such as object detection [75] and segmentation [50, 18]. Furthermore, advanced techniques like Mixup [59], StyleMix [32], and CutMix [65] have demonstrated superior performance, pushing the boundaries of data augmentation strategies. Previous studies have highlighted that the choice of augmentation strategies can significantly influence model performance, especially in contrastive learning frameworks where augmentations are crucial for generating positive and negative pairs [9]. Despite these advancements, the specific impact of various augmentations on the performance of models like SimCLR in handwritten character recognition remains underexplored, presenting an open question in the field [23].

**Our Work.** This thesis builds on prior work by applying SimCLR to the task of Greek letter recognition and comparing its performance with traditional models trained using cross-entropy and triplet loss functions. By systematically evaluating the impact of different augmentation strategies [5] on SimCLR’s performance, this research provides new insights into the strengths and limitations of contrastive learning in the context of handwritten character recognition [56]. The findings contribute to a deeper understanding of how contrastive learning models, particularly SimCLR, perform with specialized datasets [56, 60] and tasks that involve subtle distinctions between characters. In the following chapter, we will elaborate on different data augmentations and training methods, including SimCLR, for the task of letter recognition.

## Chapter 3

# Methodology

In this chapter, we shall first explore and provide a detailed discussion on the various data augmentations that were investigated in the context of Greek-letter recognition in Section 3.1. Following this, in Section 3.2, we will elaborate comprehensively on the three distinct approaches that were examined during the course of this study. Ultimately, this chapter details the training pipeline of the core methods that form the foundation for the experiments in the following chapters. Let us now first study the diverse data augmentations in detail.

### 3.1 Primary Augmentations

Data augmentation [57, 23, 75] is one of the key components in modern machine learning algorithms [6, 31]. Indeed, several simple augmentation strategies [8, 76] have led to significant improvements in results, particularly for tasks such as ImageNet classification [41]. Moreover, in the specific context of contrastive learning [36, 51, 62, 66, 71], the algorithm relies on learning from two cropped views of the same image. Therefore, our initial goal is to conduct a comprehensive study of different data augmentation strategies that will be employed in conjunction with various training methods.

In this section, we will discuss the primary augmentations that were explored for training the Greek-letter recognition models. These primary augmentations will be combined to form complex, higher-order data augmentation strategies. To achieve this, we utilize the Albumentations [5] library and conduct experiments on ten primary augmentations, comprising six spatial and four pixel-level augmentations. We will now discuss each augmentation in detail to provide a deeper understanding and to highlight the associated hyperparameters. Pixel-level augmentations modify the image at each pixel independently. In contrast, spatial-level transforms modify the input image at a global level, simultaneously affecting the entire image and any associated targets.

**Spatial Augmentations.** We consider six spatial augmentations for our analysis. Spatial augmentations are a key step in the preprocessing pipeline of the training methods.

- **Resize256.** The resize transformation resizes the input of any given image to a fixed dimension of  $256 \times 256$  pixels (shown in image 3.1a). This ensures uniformity in image dimensions across the dataset, where images typically exhibit diverse resolutions. Performing this step is crucial for training neural networks [69, 41, 48], which often require a consistent input size. By applying this transformation, we standardize the input data, making it compatible with

model requirements [6, 31] and also improving training efficiency. Each input image in the batch undergoes this resizing process to maintain standardized input dimensions. It is important to note that we perform this data augmentation step on all images during both the training and evaluation phases, and across all training methods.

- **Randomcrop224.** The random crop transformation extracts a  $224 \times 224$  pixel region from the  $256 \times 256$  image at a random location. This introduces variability in the training images by focusing on different sections of each image, which helps the model to better generalize and handle variations in object positioning. This technique is particularly useful for simulating different views of the letter images during training, ensuring that the model can recognize letters irrespective of their location. As illustrated in Figure 3.1b, a 16-image batch is randomly cropped to a size of  $224 \times 224$  pixels, effectively cropping 76% of the original image area. It is important to note that this augmentation step is highly sensitive, as cropping too small a region can shift the semantic meaning of the letter to a different label, while cropping too large a region makes the augmentation trivial.
- **Erosion.** This is a widely used augmentation in the context of letter images. Morphological erosion expands the size of objects in the image by adding pixels to their boundaries using a  $7 \times 7$  kernel. This operation enhances the visibility of features by making objects more pronounced, which can improve object detection and recognition. It is particularly useful for making features more distinct and easier for the model to detect, especially in cases where objects have thin or irregular boundaries. We can observe the effect of erosion on a 16-image batch in Figure 3.1c.
- **Dilation.** Morphological dilation reduces the size of objects in the image by removing pixels from their boundaries using a  $7 \times 7$  kernel. This operation is helpful for eliminating small noise and artifacts by shrinking object boundaries. It improves image quality by reducing the impact of minor irregularities and focusing on the main features of the objects. The effect of dilation on a 16-image batch can be seen in Figure 3.1d.
- **Affine):** The affine (shown in image 3.1e) transformation combines shifting, scaling, and rotating the image with a shift limit of 5%, scale limit of 10%, and rotation limit of 30 degrees. This augmentation simulates various perspectives and distortions, helping the model to handle changes in object positioning, size, and orientation. By applying these transformations, the model becomes more versatile in recognizing objects under different spatial conditions.
- **Hflip.** The horizontal flip, as illustrated in Figure 3.1f, mirrors the image along the vertical axis. This augmentation introduces left-right symmetry into the dataset, helping the model become invariant to horizontal orientations. By applying this transformation with a 50% probability, the model can better generalize and recognize objects regardless of their horizontal position in the image.

**Pixel-level Augmentations.** We consider four pixel augmentations for our analysis. Note that the pixel-level augmentations are applied to each pixel in the image independently with the probability  $p$  during the training.



(A) **Resize256** augmentation. The original images of varying resolutions are resized to a consistent resolution of  $256 \times 256$  pixels.



(B) **Randomcrop224** augmentation. The images are cropped to a size of  $224 \times 224$  pixels, retaining 76% of the original image area.



(C) **Morpho Erosion** removes pixels from the boundaries of objects using a  $13 \times 13$  kernel.



(D) **Morpho Dilation** expands the boundaries of objects using a  $13 \times 13$  kernel.



(E) **Affine** transformation combines shifting (0.05), scaling(0.1), and rotating(30) the image.



(F) **Hflip** horizontally flips the images.

FIGURE 3.1: Visualizations of the **spatial augmentations** applied in our experiments. The first block shows the original resized image at a  $256 \times 256$  resolution, and the following blocks show the resulting visualizations from different augmentation strategies.

- **Colorjitter.** Color jitter (shown in image 3.2b) adjusts the brightness, contrast, saturation, and hue of the image, with brightness and contrast varying within the range [0.8, 1], saturation within [0.8, 1], and hue within [-0.5, 0.5]. This augmentation increases the variability of color and lighting conditions in the dataset, helping the model to generalize across different environmental conditions and improving robustness against changes in color and illumination.
- **Gaussianblur.** Gaussian blur( 3.2d) applies a blurring effect to the image with a kernel size ranging from 3x3 to 7x7 pixels. This transformation smooths the image by averaging pixel values, which reduces sharpness and noise. By applying Gaussian blur, we help the model focus on more prominent features and improve its robustness to variations in image sharpness.
- **Invert.** The invert(we can see in image 3.2c) transformation flips the colors of the image, producing a negative of the original image. This enhancement emphasizes contrasts and highlights features that might otherwise be less visible. By inverting the colors, the model can learn to recognize features regardless of their color schemes, which can improve feature extraction and object detection performance.
- **Gray.** The grayscale, illustrated in image 3.2e transformation converts the image to shades of gray, removing color information and focusing solely on luminance. This reduction in color complexity helps the model to concentrate on structural and textural information. By applying this transformation, the model becomes better at analyzing and recognizing objects based on their shapes and textures rather than their colors.

**Remark.** An important point to emphasize at the end of this augmentation section is that several augmentations, such as Gaussian blur, colorjitter, etc., have various associated internal hyperparameters. We have fixed these internal parameters after a manual inspection of the images, and we cannot guarantee that these are optimal settings. A thorough analysis of the internal hyperparameters for each algorithm is deferred to future work. In Table 3.1, we summarize our list of primary augmentation types and their internal hyperparameters. We provide the exact parameters used for each augmentation in the Table 4.1 in the Experiments.

Index	Augmentation	Type	Hyperparameters
1	gray	Pixel-level	-
2	invert	Pixel-level	-
3	gaussianblur	Pixel-level	blur_limit, sigma_limit
4	colorjitter	Pixel-level	brightness, contrast, saturation, hue
5	resize256	Spatial-level	-
6	randomcrop224	Spatial-level	-
7	hflip	Spatial-level	-
8	morpho_dilation	Spatial-level	kernel
9	morpho_erosion	Spatial-level	kernel
10	affine	Spatial-level	shift_limit, scale_limit, rotate_limit

TABLE 3.1: Primary Augmentations.

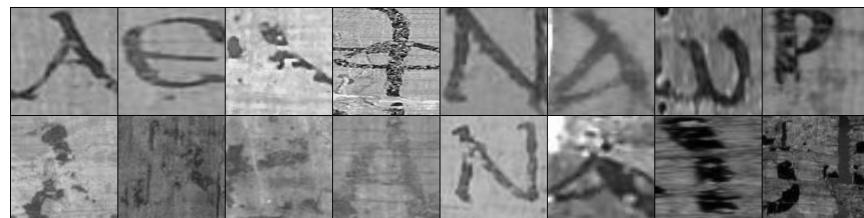
(A) **Resize256** augmentation.(B) **Colorjitter** augmentation applied to the images.(C) **Invert** augmentation applied to the images.(D) **Gaussianblur** augmentation applied to the images, removing pixels from the boundaries using a  $29 \times 29$  kernel.(E) **Grayscale** augmentation applied to convert the images to shades of gray.

FIGURE 3.2: Visualizations of the **pixel-level** augmentations applied in our experiments. The first block shows the original resized image at a  $256 \times 256$  resolution, and the following blocks show the resulting visualizations from different pixel-level augmentation strategies.

## 3.2 Training Methods

In this section, we will explain the three training methods: one baseline model and two embedding-based methods, namely Triplet and SimCLR. In short, the baseline model serves as a reference point, providing a standard against which the performance of the embedding-based methods can be compared. The Triplet method focuses on learning by comparing anchor, positive, and negative examples to enhance the model’s ability to distinguish between similar and dissimilar inputs. On the other hand, SimCLR leverages contrastive learning to maximize agreement between differently augmented views of the same data, further improving the model’s robustness and generalization capabilities at the evaluation time.

### 3.2.1 Baseline Model

We will discuss the core backbone architecture of ResNet [25] to understand the building blocks and how it encodes the input data into a final probability vector over the classes. By examining the ResNet architecture, we aim to provide a clear understanding of its structural components and the way it processes data to achieve accurate predictions. Let us now study the ResNet backbone from an architectural perspective.

**ResNet18 backbone:** We run all our experiments on the ResNet backbone [25]. ResNet18 is a CNN [69] designed for image recognition tasks and is part of the ResNet (Residual Network) family [25], which introduces residual connections between the layer inputs and outputs. These residual blocks allow the network to learn residuals, or differences, between input and output, facilitating the training of deeper architectures, and solves the problem of vanishing gradients [27] in Neural networks. The network begins with an initial convolutional layer with a  $7 \times 7$  kernel, producing an output feature map of size  $256 \times 256$  pixels, followed by a max-pooling layer that reduces the size to  $128 \times 128$  pixels. It then includes four residual stages:

- **Stage 1:** Contains 2 residual blocks, each with two  $3 \times 3$  convolutional layers (64 filters), producing an output shape of  $128 \times 128$  pixels.
- **Stage 2:** Contains 2 residual blocks, each with two  $3 \times 3$  convolutional layers (128 filters), with the output shape reduced to  $64 \times 64$  pixels.
- **Stage 3:** Contains 2 residual blocks, each with two  $3 \times 3$  convolutional layers (256 filters), with the output shape reduced to  $32 \times 32$  pixels.
- **Stage 4:** Contains 2 residual blocks, each with two  $3 \times 3$  convolutional layers (512 filters), with the output shape reduced to  $16 \times 16$  pixels.
- **Pooling:** The network concludes with a global average pooling layer that produces a feature vector of length 512. The layers up to this point are considered the backbone.
- **FC layer:** The final feature vector is passed through a fully connected layer for classification.

### Data Augmentation

In the baseline model, during training, each image is first resized to  $256 \times 256$  pixels using the `Resize256` operation to standardize the input dimensions across all dataset images. Following the resizing operation, a series of augmentations, specified in a

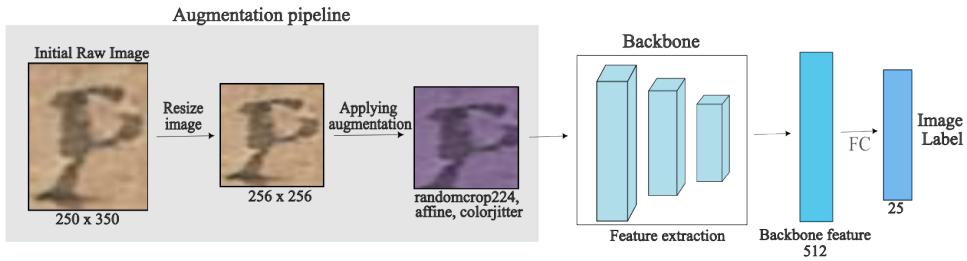


FIGURE 3.3: **Baseline model pipeline.** Spatial (eg., `randomcrop224`) and pixel-level augmentations (eg., `colorjitter`) are first performed on the original raw image to obtain the final pre-processed image. The augmented image is then passed through the model, which extracts a feature vector at the end of the backbone. The final feature vector is sent to the classification layer to produce output probabilities.

string format and split into a list of transform types, are applied. Apart from the 10 primary augmentations, a few additional transformations are applied by default to the training image. These include `CenterCrop`, which further crops the image to  $224 \times 224$  pixels, followed by the `Normalization` operation, which adjusts the image pixel values to have a mean of  $(0.485, 0.456, 0.406)$  and a standard deviation of  $(0.229, 0.224, 0.225)$ . For validation and test images, only resizing to  $256 \times 256$  pixels, center cropping to  $224 \times 224$  pixels, and normalization are applied. This ensures consistency in image dimensions and normalization while avoiding the introduction of additional variability that could affect model performance evaluation.

### Cross-Entropy Loss

We train ResNet-18 using cross-entropy loss [12] to guide the optimization process during model training. Formally, the CE loss is computed as follows:

$$L = - \sum_{i=1}^N y_i \log(\hat{y}_i) \quad (3.1)$$

where:

- $L$  is the loss value.
- $N$  is the number of classes in the training dataset.
- $y_i$  is the true label (1 if the class is the correct classification, 0 otherwise).
- $\hat{y}_i$  is the predicted probability of class  $i$ .

Cross-entropy loss evaluates the divergence between the true labels and the predicted probabilities. By penalizing higher deviations between predicted probabilities and actual labels, it ensures that the model learns to predict probabilities that are as close to the true labels as possible.

## Overall Training Pipeline

As shown in Figure 3.3, the baseline model process follows a straightforward sequence from input image to output class label:

1. Each training image is resized to a standard dimension, typically  $256 \times 256$  pixels. The resized image is then subjected to a series of augmentations, both spatial and pixel-level. We apply spatial augmentations with a probability of 1 but this only applies to `resize256` and `randomcrop224`, meaning all images in the batch undergo `randomcrop224`; the remaining primary spatial-level augmentations have a probability of 0.5. However, pixel-level augmentations such as `color-jittering`, `blurring`, and `grayscale` are applied with a probability of 0.5. This introduces variability and diversity into the training data, helping the model generalize better.
2. The augmented images are fed into ResNet-18, which extracts hierarchical features from the images through multiple layers of convolutions and residual blocks. At the end of the backbone, a globally averaged pooled feature vector is obtained. This feature vector is passed through a fully connected classification layer, which outputs class probabilities for each image. The class with the highest probability is chosen as the predicted label for that image.
3. To drive the optimization, we use the cross-entropy loss function, which evaluates the discrepancy between the predicted probabilities and the actual class labels. For each image in the batch, the cross-entropy loss is computed as shown in Equation 3.1. The cross-entropy loss for the entire batch is averaged from the losses of individual images. Using the computed loss, the optimizer( [39]) performs backpropagation to update the model’s weights.
4. The model undergoes iterative training, processing multiple batches of images. During each iteration, the model learns from the computed losses and adjusts its weights accordingly. This iterative process continues until the model’s performance stabilizes.
5. After training, the model is evaluated on validation and test datasets. Performance metrics, such as accuracy, are computed to assess how well the model generalizes to new, unseen data.

### 3.2.2 Triplet Model

The Triplet Model [31] was integrated into our experiments to complement the baseline model by learning more discriminative embeddings and to provide a comparison with the baseline and SimCLR self-supervised models. It is particularly effective for tasks requiring fine-grained distinctions between classes, such as face recognition [55, 4] and person reidentification [72, 29]. The model is trained using the triplet loss, which helps in creating compact clusters of similar examples while pushing apart dissimilar ones.

### Architecture

We use ResNet as the backbone in the experiment. The final feature vector, which is the output of the layer just before the classification layer, is extracted. This feature vector is then fed into a shallow single layer to project the feature from its original dimension (512 dimension) to 64 dimensions, improving the efficiency of the training

process. It is important to note that during the triplet pretraining stage, no classification layer is involved. However, during the fine-tuning stage, we add a classification layer on top of the projection layer to predict the label.



FIGURE 3.4: **Batch of images used in Triplet model training.**

The first row represents the **anchors**, the second row represents the **positives**, which contain images with the same label as the anchors, and the third row shows the **negatives**, which are images with different labels from the anchors.

### Triplet Loss

The Triplet Loss model was integrated into our experiments to compare with the baseline model. The Triplet Embedding model is designed to train a neural network to generate embeddings where the distance between an anchor and a positive example (from the same class) is smaller than the distance between the anchor and a negative example (from a different class) by a defined margin, which is a hyperparameter. This loss function encourages the network to create tightly packed clusters of similar items while pushing dissimilar items apart. The goal is to minimize the distance between the anchor and positive images while maximizing the distance between the anchor and negative images. This approach is particularly beneficial for tasks such as face recognition, where it is essential to differentiate between similar and dissimilar identities. Formally, we calculate the triplet loss as follows:

$$L(\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n) = \max(0, D(\mathbf{x}_a, \mathbf{x}_p) - D(\mathbf{x}_a, \mathbf{x}_n) + \alpha) \quad (3.2)$$

where:

- $\mathbf{x}_a$  is the representation of the anchor instance.
- $\mathbf{x}_p$  is the representation of the positive instance (same class as anchor).
- $\mathbf{x}_n$  is the representation of the negative instance (different class from anchor).
- $\alpha$  is the margin, a positive constant that ensures a minimum separation between positive and negative pairs.
- $D(\mathbf{x}_a, \mathbf{x}_p)$  is the distance metric (often Euclidean distance) between the embeddings of  $\mathbf{x}_a$  and  $\mathbf{x}_p$ , which should be minimized.
- $D(\mathbf{x}_a, \mathbf{x}_n)$  is the distance between the anchor and negative examples, which should be maximized.

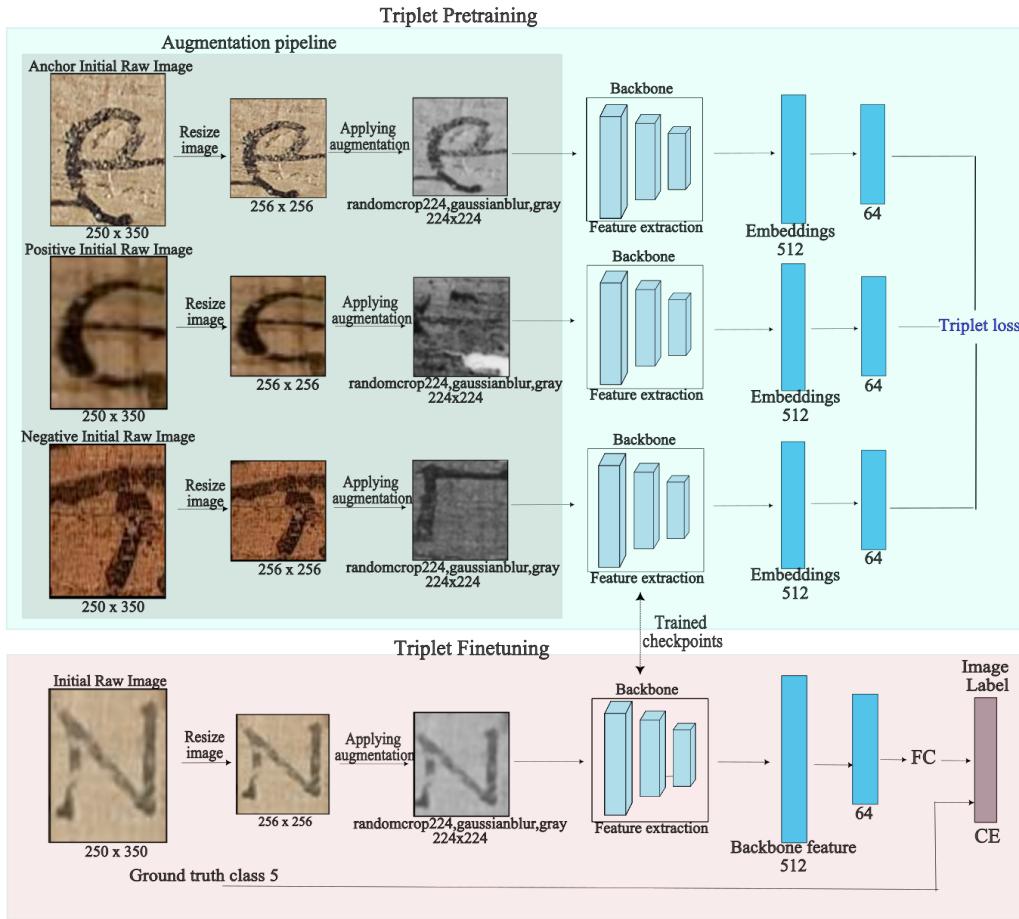


FIGURE 3.5: **Triplet Model Pipeline.** The top block, highlighted with a green background, represents the pretraining stage, where the model is trained on the pretraining dataset using triplet loss to learn embeddings from triplet pairs. In the next stage, a classification layer is added on top of the embedding layer, and the model is trained end-to-end with cross-entropy (CE) loss.

The term  $\max(0, D(\mathbf{x}_a, \mathbf{x}_p) - D(\mathbf{x}_a, \mathbf{x}_n) + \alpha)$  ensures that the loss is computed only when the distance between the anchor and the positive instance is not at least the margin  $\alpha$  smaller than the distance between the anchor and the negative instance. In other words, the loss is zero if the positive pair is at least  $\alpha$  closer to the anchor compared to the negative pair.

### Data Augmentation

This follows a similar theme as in the baseline model. Here, we first resize images to  $256 \times 256$  pixels and apply spatial augmentation by randomly cropping a  $224 \times 224$  region and series of augmentations, followed by pixel-level augmentations such as **CenterCrop**, **Normalization**, etc. For validation and test images, we apply spatial-level augmentation like resizing to  $256 \times 256$  pixels, then apply **CenterCrop** to  $224 \times 224$  pixels, followed by **Normalization**. We do not apply pixel augmentations during the testing phase.

In triplet model training, the dataloader extracts a batch of  $N$  original Greek letter images, randomly selected from the training set. These images are considered as anchors. The dataloader also extracts another  $N$  images from the same class as the anchor images, which are considered positives, and another set of  $N$  images that have different labels from the original images, which are considered negatives. Overall, the resulting batch contains  $N \times 3$  images (see Figure 3.4 for an example with a batch of 8 images). In short, each triplet consists of an anchor image, a positive image (another crop of the same original image as the anchor), and a negative image (a crop from a different original image).

## Overall Training Pipeline

### 1. Pretraining Stage

- (a) As shown in the Figure 3.5 top block denoted within a green background, the triplet model operates based on three types of samples: anchor, positive, and negative. The model starts with a batch of images. Each batch contains multiple triplets, where each triplet is composed of three images: an anchor image, a positive image, and a negative image. For instance, if the batch size is 16, it will consist of several triplets where each triplet contains three different images. Before feeding the images into the model, they undergo a series of augmentations, such as cropping, color jittering, and blurring. These augmentations help in creating diverse views of the same image and improve the robustness of the learned embeddings.
- (b) Each image in the batch (anchor, positive, and negative) is passed through a neural network, such as ResNet-18 or ResNet-50. This forward pass generates embeddings, which are vector representations of the images in a high-dimensional space. After passing through the network, each image is represented by an embedding vector. The embeddings for the anchor, positive, and negative images are obtained from the output of the network.
- (c) The triplet loss function (Equation 3.2) calculates the distances between the anchor-positive and anchor-negative embeddings. Typically, the Euclidean distance or cosine similarity is used to measure these distances. The loss function enforces that the distance between the anchor and positive embeddings should be smaller than the distance between the anchor and negative embeddings by at least  $\alpha$ . If the distance condition is met, the loss is zero; otherwise, it penalizes the model based on how much the condition is violated.
- (d) The computed loss is used to calculate gradients with respect to the model parameters. This involves backpropagation, where the gradients are propagated backward through the network to update the weights. The optimizer (e.g., Adam [39] or SGD [53]) updates the network weights based on the gradients. This process adjusts the embeddings so that similar images (anchor and positive) are closer together, and dissimilar images (anchor and negative) are pushed further apart in the embedding space.

### 2. Fine-tuning Stage

- (a) Once the embedding model is pre-trained using the triplet loss, it can be fine-tuned on a smaller, more specific dataset. During fine-tuning, the pre-trained embeddings are used as a starting point, and the model is further

trained to adapt to the new dataset’s class distribution. To this end, a classification layer is added on top of the embeddings to predict class labels. This layer uses the embeddings from the backbone to classify images into one of the predefined classes based on their proximity in the embedding space. We train this model end-to-end using the CE loss. Optionally, we can freeze the backbone network to minimize computational costs.

- (b) Note that during this stage, we use the same augmentation that was applied during the pretraining stage.

### 3.2.3 SimCLR

We now discuss the third training method, SimCLR [6], the main focus of our thesis. SimCLR [6] is a self-supervised learning framework and one of the popular contrastive learning techniques in machine learning [51, 36, 62, 66]. Unlike the baseline method in Section 3.2.1 and triplet training method in Section 3.2.2 which requires ground truth labels, SimCLR pretraining does not require image labels and hence can be pretrained on a massive corpus of data without expensive annotations. The goal of SimCLR is to learn meaningful embeddings from the raw images. To this end, the SimCLR model takes augmented views, i.e., cropped subimages, of the same image and pulls the embeddings of these augmented images closer in the latent space, while the embeddings coming from two different images are pushed apart. We illustrate the pipeline of SimCLR in Figure 3.7, which contains the details about the pretraining on large corpus dataset in the top block and the details about fine-tuning in the bottom block. We will now discuss the architecture of SimCLR below:

**Architecture.** The SimCLR architecture follows the same theme as the Triplet model architecture, where we focus on training embedding than the labels. Here is the overview:

- **Backbone:** Typically based on ResNet, producing a feature vector of dimension  $D$  which is 512 for ResNet18 [25].
- **Projection Head:** A multi-layer perceptron (MLP) that maps the backbone’s feature vector (e.g., 512 dimensions in ResNet18 [25]) to a lower-dimensional space (e.g., 128 dimensions).
- **Contrastive Loss Function:** The model is trained using a InfoNCE loss function that pulls the embeddings of crops from the same image closer together while pushing apart embeddings of different images.

**InfoNCE Loss:** In SimCLR, we use InfoNCE loss [3, 19, 6] (Information Noise Contrastive Estimation) to train the model. InfoNCE loss is designed to maximize the mutual information between different views or augmentations of the same data instance while minimizing the similarity between different data instances. This loss function leverages the concept of contrasting positive samples (augmentations of the same instance) against a set of negative samples (other instances). By doing so, it encourages the model to learn representations that bring similar instances closer together in the embedding space while pushing dissimilar instances apart. In short, each image is assigned a label index corresponding to the position where the sister image, originating from the same parent image, is present in the batch. The feature is represented based on the normalized distance between the given image and all the

other images in the batch.

Formally, the InfoNCE loss is calculated as follows:

$$L = - \sum_{i=1}^N \log \left( \frac{\exp\left(\frac{\text{sim}(\mathbf{x}_i, \mathbf{x}_i^+)}{\tau}\right)}{\sum_{j=1}^{2N} \mathbf{1}_{[j \neq i]} \exp\left(\frac{\text{sim}(\mathbf{x}_i, \mathbf{x}_j)}{\tau}\right)} \right) \quad (3.3)$$

where:

1.  $L$  is the loss.
2.  $N$  is the number of positive pairs.
3.  $\mathbf{x}_i$  and  $\mathbf{x}_i^+$  are the representations of the anchor and positive examples, respectively.
4.  $\text{sim}(\mathbf{x}_i, \mathbf{x}_j)$  is the similarity function (often the dot product or cosine similarity).
5.  $\tau$  is a temperature parameter.
6.  $\mathbf{1}_{[j \neq i]}$  is an indicator function that is 1 if  $j \neq i$ , and 0 otherwise.

## Data Augmentations

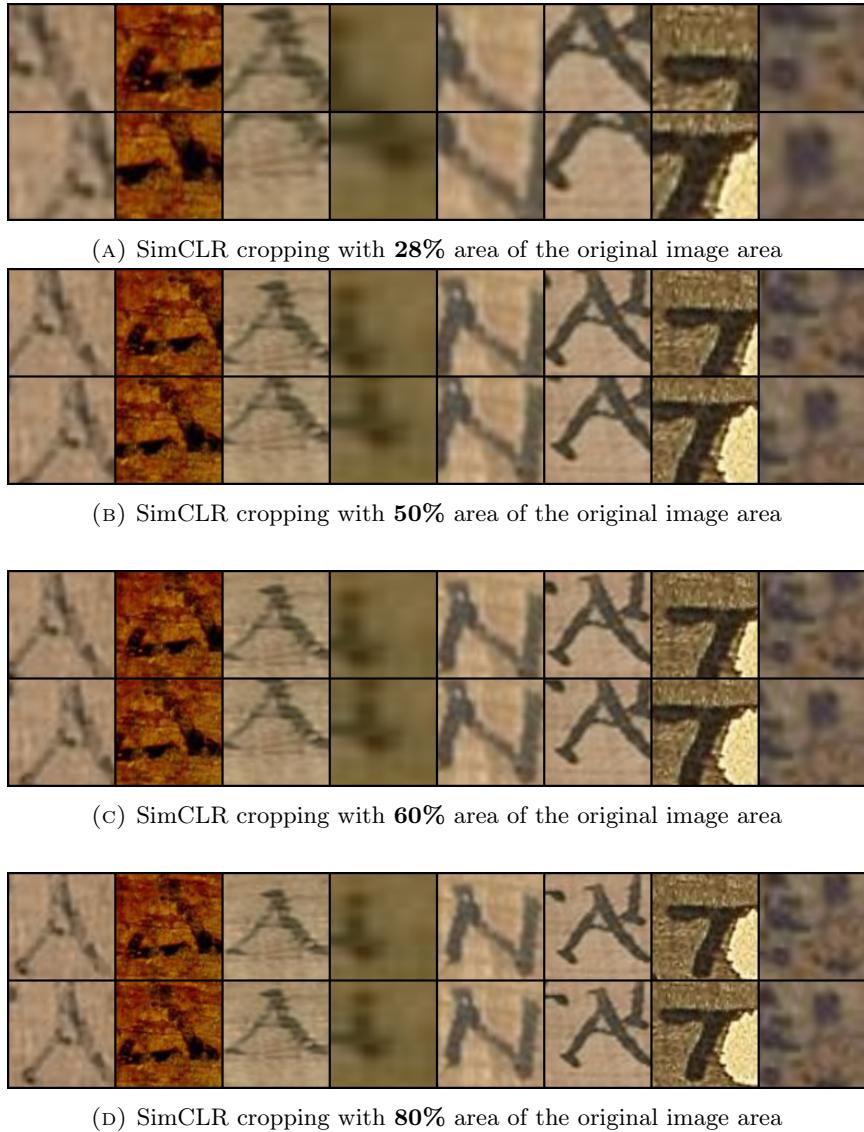
In the SimCLR model, images are initially resized to  $256 \times 256$  pixels to standardize input dimensions, similar to earlier methods. Following this, a sequence of augmentations, including spatial and pixel-level augmentations specified as a list of transform types, is applied. After applying these augmentations, we resize the image to  $96 \times 96$  pixels. This step is necessary because SimCLR requires larger batch sizes for convergence. However, due to memory constraints, we perform resizing to  $96 \times 96$  pixels to manage memory usage effectively while still maintaining performance.

An important aspect when performing augmentations is the cropping size. After resizing the original image to  $256 \times 256$  pixels, we apply random cropping to ensure that at least 60% of the original image is visible during augmentation. In Figure 3.6c, we observe a batch of 8 images with 60% visibility, where the model evaluates images where the letter occupies at least 60% of the original image area, corresponding to a size of  $198 \times 198$  pixels. Additional examples of cropping with 50%, 28%, and 80% visibility can be seen in Figures 3.6b, 3.6a, and 3.6d, respectively.

**Remark.** While it is challenging to determine the optimal cropping size, due to limited resources, we agreed to use 60% of the original image after multiple rounds of discussion with the professor.

## Overall Training Pipeline

Typically, SimCLR is trained in two stages. In the first stage, embeddings are learned on a large corpus of unlabeled data. In the second stage, supervised training is conducted on top of the embeddings learned from the first stage using an additional classification layer. It is important to note that one can also choose other methods, such as k-Nearest Neighbors (kNN), in the second stage. Let us now discuss the two stages in detail:



**FIGURE 3.6: Visualizations of cropping image batch with varying areas.** We observe that cropping areas limited to 28% altered the image substantially, whereas the other three areas, such as 50%, 60%, and 80%, appear more reasonable. After consensus with the project team, we chose the 60% area for our main experiments.

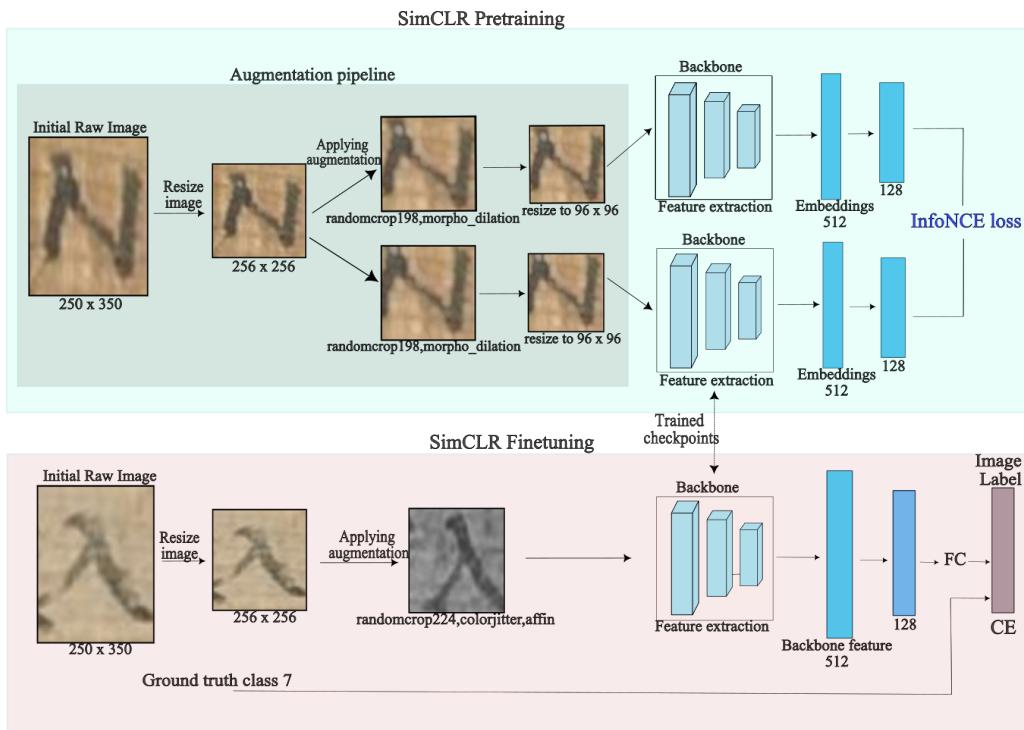


FIGURE 3.7: **SimCLR Pipeline.** The top block, highlighted with a green background, represents the pretraining stage, where the model is trained on the pretraining dataset using InfoNCE loss to learn embeddings from SimCLR augmented views of the same image. In the next stage, a classification layer is added on top of the embedding layer, and the model is trained end-to-end with cross-entropy (CE) loss.

### 1. Pretraining Stage

- (a) In this stage, we train the network in a self-supervised manner on a large-scale dataset without the need for ground truth labels. The algorithm first resizes the original images to a uniform size of  $256 \times 256$  pixels. Following this, each image is cropped to  $198 \times 198$  pixels and then subjected to various augmentations, including further cropping, color jittering, and blurring, to create **two** diverse views of each image.
- (b) The two cropped views form the positive pair and are then passed through the model to obtain the positive embeddings. The other images in the batch form the negative images, which are then passed through the model to get the negative embeddings.
- (c) We optimize the backbone network by minimizing the InfoNCE loss, ultimately learning a rich embedding space.

### 2. Fine-tuning Stage

- (a) In the fine-tuning stage, we use the pre-trained backbone from the earlier pretraining stage and then fine-tune it on a small labeled dataset such as ICDAR [56] in our setting.
- (b) To perform supervised learning, we add a classification layer on top of the SimCLR backbone to get the probability distribution over the classes. This is a simple fully connected layer.
- (c) We train the full network, including the backbone, using cross-entropy loss. We will also show in the experiments that the full training of the backbone is essential since the SimCLR embeddings are not quite discriminative even after training on the large corpus.

Overall, the SimCLR training leverages large publicly available datasets to learn rich embedding representations, which are then transferred to the target fine-tuning domain with small-scale supervised training.

## 3.3 Summary

In this chapter, we first studied the data augmentations that were explored for our experiments on the task of Greek-letter recognition. Specifically, we selected ten augmentations, comprising six spatial and four pixel-level augmentations. These augmentations were tailored to each training method.

Next, we discussed three training methods: the baseline model using cross-entropy loss, the triplet model with triplet loss, and the SimCLR model with InfoNCE loss. While the baseline model learns directly from the labels, the other two models learn a rich embedding space based on the images in the batch. Moreover, the triplet model requires labels to prepare the training batch, whereas SimCLR pretraining does not depend on labels. In the second stage, both the triplet and SimCLR models undergo supervised fine-tuning by adding an additional classification layer. We will present the key results, highlighting the effectiveness of the augmentations and training methods, in the following chapter.

## Chapter 4

# Experiments

In this chapter, we present the experimental results of Greek papyrus letter recognition using different methods. We begin by introducing the datasets and detailing the key experimental settings. Following this, we provide quantitative results for each method, supported by t-SNE [63] visualizations of the embeddings.

### 4.1 Experimental Setting

We provide a detailed discussion of the key datasets, implementation details, training-time hyperparameters, and the various data augmentation strategies employed in our experiments. To begin, let's first examine the datasets used in this study.

#### 4.1.1 Datasets

We conduct our experiments using two datasets: ALPUB [60] and ICDAR [56].

- 1. Pretraining Dataset.** We use the ALPUB [60] dataset for the pretraining stage. This dataset comprises a comprehensive collection of ancient Greek papyrus fragments, featuring a wide variety of handwritten letters. It contains 24 distinct letter classes, making it an invaluable resource for pretraining unsupervised models for Greek character recognition. The dataset includes a total of 205,797 cropped Greek letter images. We utilize this dataset in the pretraining phase for the Baseline [25, 12], Triplet [31], and SimCLR [31] embedding models.
- 2. Finetuning Dataset.** The ICDAR dataset is used for finetuning. This dataset, which was part of the ICDAR 2023 competition focused on the detection and recognition of Greek letters on papyri [56], consists of 34,061 cropped images. We split the dataset into training, validation, and testing sets in a 70%, 15%, and 15% ratio, resulting in 23,842 images for training, 5,109 for validation, and 5,110 for testing. It is important to note that the ICDAR dataset [56] includes an additional class compared to the ALPUB dataset [60], bringing the total to 25 classes. Furthermore, the original dataset provided full images, which we cropped ourselves using ground-truth annotations provided by the authors.

#### 4.1.2 Implementation Details

Our experiments were conducted on a setup consisting of four NVIDIA GPUs, each with 9GB of RAM. For hyperparameter tuning, we utilized RayTune [43], a robust library that efficiently explores the hyperparameter space. The experiments were implemented using the PyTorch library [52], which provides a flexible framework for

deep learning tasks. Additionally, we leveraged the Albumentations library [5] to generate a diverse set of augmentations, ensuring that our models were exposed to a wide range of data variations. Below, we discuss the key experimental parameters used at different stages for each method.

**Pretraining Stage.** During the pretraining phase, we used the Adam optimizer [39] with a learning rate of 0.001 for both the Triplet embedding model and the Baseline model. A learning rate scheduler with a gamma of 0.1 was employed, and the models were trained for 20 epochs and temperature values used for triplet 1.0. For training the SimCLR embedding model, we also used the Adam optimizer [39], but with a slightly lower learning rate of 0.0003, and temperature value is 0.07. The SimCLR model was trained for 100 epochs, utilizing a Cosine Annealing scheduler [45] to gradually reduce the learning rate throughout the training process. This careful adjustment of the learning rate was intended to help the model converge more effectively.

**Finetuning Stage.** In the finetuning stage, we transitioned to training our models on the ICDAR dataset [56]. For this phase, we added a classification layer to the embedding models (Triplet and SimCLR) and finetuned them by either adjusting only the last layer or the entire model. All models during this stage were trained using the Adam optimizer [39] with a learning rate of 0.001, and a learning rate scheduler was employed over 20 epochs. Due to the constraints of our computational resources, we did not conduct extensive tuning of the optimizer hyperparameters, opting instead to focus on ensuring that the models were sufficiently trained under the given conditions.

#### 4.1.3 Methods

In this study, we evaluate the performance of Greek letter recognition using three different training strategies during the pretraining stage:

1. **Baseline Model:** The baseline model is trained using the Cross-entropy loss [12] function, which relies on ground-truth labels to guide the learning process. This method serves as a standard supervised learning approach. Detailed training procedures for this method are provided in Section 3.2.1.
2. **Triplet Embedding Model:** This model utilizes the Triplet loss function, where triplets of images (anchor, positive, and negative) are selected based on ground-truth labels to learn discriminative embeddings. As a result, the pre-training phase for this model requires access to labeled data from the pretraining dataset. Complete training details for this method are available in Section 3.2.2.
3. **SimCLR Model:** The SimCLR model is trained using the InfoNCE loss in a completely unsupervised manner, meaning it does not require access to image labels. This method leverages contrastive learning to learn robust feature representations. We outline the full training details for this method in Section 3.2.3.

In the finetuning stage, we add an additional classification layer to all the models and retrain them using Cross-entropy loss. This final step allows us to benchmark the performance of each model on the ICDAR dataset, providing a fair comparison across different training strategies.

#### 4.1.4 Data Augmentations

Our experiments utilize a pool of 93 augmentations, consisting of various combinations of primary augmentation techniques, as discussed in Section 3.1. We create augmentation pipelines by combining two, three, or four primary augmentations for each image. For experiments conducted without pretraining on the large-scale ALPUB dataset, we evaluated all 93 augmentations. However, when pretraining on ALPUB was involved, we focused on the top-4 augmentations identified from our initial experiments without pretraining. During training, both spatial augmentations and pixel-level augmentations are applied to enhance model generalization. For testing, we report the results without applying any additional augmentations, ensuring that the evaluation reflects the model’s performance on unaltered data. We provide the exact values for hyperparameters in Table 4.1. Note that, we selected these values after careful manual inspection so that the augmented image retains same semantic label meaning.

Index	Augmentation	Type	Hyperparameters
1	invert	Pixel-level	-
2	gray	Pixel-level	-
3	gaussianblur	Pixel-level	blur limit = (3, 7), sigma limit = 0
4	colorjitter	Pixel-level	brightness = (0.8, 1), saturation = (0.8, 1), contrast = (0.8, 1), hue = (-0.5, 0.5)
5	resize256	Spatial-level	-
6	randomcrop224	Spatial-level	-
7	hflip	Spatial-level	-
8	morpho_dilation	Spatial-level	kernel(w, h) = (7, 7)
9	morpho_erosion	Spatial-level	kernel(w, h) = (7, 7)
10	affine	Spatial-level	shift_limit = 0.05, scale_limit = 0.1. rotate_limit = 30

TABLE 4.1: Hyperparameters of Primary Augmentations

- **First Order Combinations.** Each combination includes only one augmentation applied to the base "randomcrop224". This augmentation test the effect of individual augmentations on the dataset. It serves as a baseline with no additional augmentations applied. We show the augmentation in Table 4.2 for the sake of the clarity.
- **Second Order Combinations.** Table 4.3 lists all possible combinations of two augmentations applied to randomcrop224. This is useful for understanding how pairs of augmentations interact with each other.
- **Third Order Combinations** Table 4.4 covers all combinations of three augmentations. This helps in examining how combinations of three augmentations affect the dataset and interact with each other.

- **Fourth Order Combinations** This table 4.5 includes all combinations of four augmentations applied together. It helps to understand the combined effect of multiple augmentations and their interactions.

Index	Augmentation
1	randomcrop224

TABLE 4.2: **First-Order Augmentation.** This baseline augmentation randomly crops a  $224 \times 224$  region from a resized  $256 \times 256$  image.

Index	Augmentation
1	randomcrop224,morpho\_erosion
2	randomcrop224,morpho\_dilation
3	randomcrop224,affine
4	randomcrop224,colorjitter
5	randomcrop224,hflip
6	randomcrop224,invert
7	randomcrop224,gaussianblur
8	randomcrop224,gray

TABLE 4.3: **Second-Order Combinations.** This augmentation first performs the baseline step of randomly cropping a  $224 \times 224$  region from a resized  $256 \times 256$  image, followed by the application of additional augmentations.

Index	Augmentation
1	randomcrop224,morpho\_erosion,morpho\_dilation
2	randomcrop224,morpho\_erosion,affine
3	randomcrop224,morpho\_erosion,colorjitter
4	randomcrop224,morpho\_erosion,hflip
5	randomcrop224,morpho\_erosion,invert
6	randomcrop224,morpho\_erosion,gaussianblur
7	randomcrop224,morpho\_erosion,gray
8	randomcrop224,morpho\_dilation,affine
9	randomcrop224,morpho\_dilation,colorjitter
10	randomcrop224,morpho\_dilation,hflip
11	randomcrop224,morpho\_dilation,invert
12	randomcrop224,morpho\_dilation,gaussianblur
13	randomcrop224,morpho\_dilation,gray
14	randomcrop224,affine,colorjitter
15	randomcrop224,affine,hflip
16	randomcrop224,affine,invert
17	randomcrop224,affine,gaussianblur
18	randomcrop224,affine,gray
19	randomcrop224,colorjitter,hflip
20	randomcrop224,colorjitter,invert
21	randomcrop224,colorjitter,gaussianblur
22	randomcrop224,colorjitter,gray

Index	Augmentation
23	randomcrop224,hflip,invert
24	randomcrop224,hflip,gaussianblur
25	randomcrop224,hflip,gray
26	randomcrop224,invert,gaussianblur
27	randomcrop224,invert,gray
28	randomcrop224,gaussianblur,gray

TABLE 4.4: **Third-Order Combinations.** This augmentation begins with the baseline step of randomly cropping a  $224 \times 224$  region from a resized  $256 \times 256$  image, followed by the application of two additional augmentations.

Index	Augmentation
1	randomcrop224,morpho\_erosion,morpho\_dilation,affine
2	randomcrop224,morpho\_erosion,morpho\_dilation,colorjitter
3	randomcrop224,morpho\_erosion,morpho\_dilation,hflip
4	randomcrop224,morpho\_erosion,morpho\_dilation,invert
5	randomcrop224,morpho\_erosion,morpho\_dilation,gaussianblur
6	randomcrop224,morpho\_erosion,morpho\_dilation,gray
7	randomcrop224,morpho\_erosion,affine,colorjitter
8	randomcrop224,morpho\_erosion,affine,hflip
9	randomcrop224,morpho\_erosion,affine,invert
10	randomcrop224,morpho\_erosion,affine,gaussianblur
11	randomcrop224,morpho\_erosion,affine,gray
12	randomcrop224,morpho\_erosion,colorjitter,hflip
13	randomcrop224,morpho\_erosion,colorjitter,invert
14	randomcrop224,morpho\_erosion,colorjitter,gaussianblur
15	randomcrop224,morpho\_erosion,colorjitter,gray
16	randomcrop224,morpho\_erosion,hflip,invert
17	randomcrop224,morpho\_erosion,hflip,gaussianblur
18	randomcrop224,morpho\_erosion,hflip,gray
19	randomcrop224,morpho\_erosion,invert,gaussianblur
20	randomcrop224,morpho\_erosion,invert,gray
21	randomcrop224,morpho\_erosion,gaussianblur,gray
22	randomcrop224,morpho\_dilation,affine,colorjitter
23	randomcrop224,morpho\_dilation,affine,hflip
24	randomcrop224,morpho\_dilation,affine,invert
25	randomcrop224,morpho\_dilation,affine,gaussianblur
26	randomcrop224,morpho\_dilation,affine,gray
27	randomcrop224,morpho\_dilation,colorjitter,hflip
28	randomcrop224,morpho\_dilation,colorjitter,invert
29	randomcrop224,morpho\_dilation,colorjitter,gaussianblur
30	randomcrop224,morpho\_dilation,colorjitter,gray
31	randomcrop224,morpho\_dilation,hflip,invert
32	randomcrop224,morpho\_dilation,hflip,gaussianblur
33	randomcrop224,morpho\_dilation,hflip,gray
34	randomcrop224,morpho\_dilation,invert,gaussianblur
35	randomcrop224,morpho\_dilation,invert,gray
36	randomcrop224,morpho\_dilation,gaussianblur,gray

Index	Augmentation
37	randomcrop224,affine,colorjitter,hflip
38	randomcrop224,affine,colorjitter,invert
39	randomcrop224,affine,colorjitter,gaussianblur
40	randomcrop224,affine,colorjitter,gray
41	randomcrop224,affine,hflip,invert
42	randomcrop224,affine,hflip,gaussianblur
43	randomcrop224,affine,hflip,gray
44	randomcrop224,affine,invert,gaussianblur
45	randomcrop224,affine,invert,gray
46	randomcrop224,affine,gaussianblur,gray
47	randomcrop224,colorjitter,hflip,invert
48	randomcrop224,colorjitter,hflip,gaussianblur
49	randomcrop224,colorjitter,hflip,gray
50	randomcrop224,colorjitter,invert,gaussianblur
51	randomcrop224,colorjitter,invert,gray
52	randomcrop224,colorjitter,gaussianblur,gray
53	randomcrop224,hflip,invert,gaussianblur
54	randomcrop224,hflip,invert,gray
55	randomcrop224,hflip,gaussianblur,gray
56	randomcrop224,invert,gaussianblur,gray

TABLE 4.5: **Fourth-Order Combinations.** This augmentation starts with the baseline step of randomly cropping a  $224 \times 224$  region from a resized  $256 \times 256$  image, followed by the application of three additional augmentations.

## 4.2 Results

In this section, we present and discuss the experimental results obtained from the three different methods explored in this study. Our analysis is divided into two main parts to provide a clear comparison of the impact of pretraining on the large-scale ALPUB dataset.

We first delve into the results obtained without any pretraining on the ALPUB dataset, which are discussed in detail in Section 4.2.1. This analysis allows us to understand how each method performs when trained directly on the ICDAR dataset without leveraging additional pretraining data. The performance of these models in this setting provides a baseline for comparison, highlighting the strengths and limitations of each approach when operating in a more constrained environment with potentially less data diversity.

Subsequently, in Section 4.2.2, we shift our focus to the results achieved after pretraining on the ALPUB dataset. This phase involves first pretraining the models on the extensive ALPUB dataset to learn robust feature representations, followed by finetuning on the more specific ICDAR dataset. By comparing these results with those from the previous section, we aim to elucidate the benefits and potential improvements that pretraining on a large and diverse dataset like ALPUB can bring to the task of Greek letter recognition.

Overall, this two-part analysis will provide a thorough understanding of how pretraining influences model performance and help identify the most effective strategies for enhancing Greek letter recognition accuracy.

### 4.2.1 Results without Pretraining on Alpub

In this section, we present the results obtained from direct training on the ICDAR dataset [56] using the three methods under consideration: the Baseline model, the Triplet Embedding model, and the SimCLR model. The results for ResNet-18 [25] and ResNet-50 [25] architectures are summarized in Table 3.2.1, 3.2.2 and Table 3.2.3, respectively.

We conducted experiments using 93 different data augmentations (as detailed in Tables 4.2, 4.3, 4.4, 4.5) for each method. We distill the key results and report only the best-performing augmentation for each method in this section, and defer the complete results to the Tables(A.1, A.3, A.9, A.12, A.24, A.21) and (A.2, A.5, A.15, A.18 A.30, A.27) in the Appendix.

**Key finding.** Our findings reveal that the **Baseline model** trained with cross-entropy loss consistently achieved the highest test accuracy, reaching **80.67%** on ResNet-18 [25], and slightly lower at **80.47%** on ResNet-50 [25]. These results indicate that the cross-entropy-based Baseline model outperformed both the embedding-based methods across both architectures.

In comparison, the Triplet and SimCLR models achieved lower accuracies, with the Triplet model reaching 78.22% and the SimCLR model achieving 79.24% on ResNet-50. This trend highlights the robustness of the Baseline model in this direct training scenario. However, it's important to note that the best-performing augmentation strategy varied between methods, with the more complex fourth-order augmentations often leading to better performance. Specifically, the augmentation pipeline involving `randomcrop224`, `morpho_erosion`, `morpho_dilation`, `gaussianblur` stood out, achieving the highest accuracy in two out of the six experiments (ie., six rows) conducted across the two architectures.

These results underscore the significance of selecting the appropriate combination of augmentations tailored to each method. The success of more advanced augmentation strategies, such as third and fourth-order combinations that integrate multiple morphological transformations and pixel-level operations, suggests that careful tuning is essential for maximizing model performance. The diversity and complexity of these augmentations appear to be particularly beneficial, likely by enhancing the model's ability to generalize across varied data conditions. Overall, these findings highlight the need for a nuanced approach to data augmentation, where the choice of strategy is method-specific and can significantly influence the outcome of the training process.

### 4.2.2 Results with Pretraining

This section discusses the core results that are central to this thesis: specifically, how pretraining on a large-scale dataset impacts the performance of contrastive learning techniques. We will address this question with empirical evidence, analyzing the effects of pretraining on model performance in the following paragraphs.

Experiment	Dataset	Best augmentation	Valid Acc.	Test Acc.
Baseline model	ICDAR	randomcrop224, morpho_erosion, morpho_dilation, gaussianblur	81.19%	<b>80.67%</b>
Triplet model	ICDAR	randomcrop224, morpho_dilation, affine, colorjitter	80.11%	79.16%
SimCLR model	ICDAR	randomcrop224, affine, colorjitter, gray	80.33%	80.00%

TABLE 4.6: **Results on ResNet-18 without pretraining on Alpub dataset.** We report the best found augmentation and their corresponding validation and test set accuracies by directly finetuning on ICDAR. We observe the baseline model achieves the best results than other two methods.

Experiment	Dataset	Best augmentation	Valid Acc.	Test Acc.
Baseline model	ICDAR	randomcrop224, morpho_erosion, gaussianblur	80.70%	<b>80.47%</b>
Triplet model	ICDAR	randomcrop224, morpho_erosion, morpho_dilation, gaussianblur	79.29%	78.22%
SimCLR model	ICDAR	randomcrop224, colorjitter, gaussianblur	80.05%	79.24%

TABLE 4.7: **Results on ResNet-50 without pretraining on Alpub dataset.** We report the best found augmentation and their corresponding validation and test set accuracies. We observe the baseline model achieves the best results than other two methods.

**Selecting Top-4 Augmentations for Pretraining on the ALPUB Dataset.** Before presenting the results of pretraining on the ALPUB dataset using different augmentations, we encountered significant computational bottlenecks. Given these constraints of limited computational resources, we selected only a few top-performing augmentations from our earlier experiment, which involved direct finetuning on the ICDAR dataset using SimCLR. We perform top-4 selection with two strategies.

1. **Strategy 1: T-test based selection.** To ensure the effectiveness of our selection, we conducted a statistical analysis to identify the top four augmentations. This process involved running the SimCLR method across 93 different augmentations, using three random seeds for each. Through a paired t-test, we

Experiment	Dataset	Best augmentation	Valid Acc.	Test Acc.
Baseline model	Alpub + ICDAR	randomcrop224, hflip, gray	80.49%	<b>79.94%</b>
Triplet model	Alpub + ICDAR	randomcrop224, morpho_dilation, hflip	78.19%	77.51%
SimCLR model	Alpub + ICDAR	randomcrop224, colorjitter, hflip, invert	77.55%	76.14%

TABLE 4.8: **Results on ResNet-18 with pretraining on Alpub dataset (with top-4 selected using strategy 1).** We report the best found augmentation and their corresponding validation and test set accuracies. We observe the baseline model achieves the best results than other two methods.

Experiment	Dataset	Best augmentation	Valid Acc.	Test Acc.
Baseline model	Alpub + ICDAR	randomcrop224, morpho_dilation, hflip	80.21%	<b>79.75%</b>
Triplet model	Alpub + ICDAR	randomcrop224, invert, gaussianblur, gray	77.90%	77.03%
SimCLR model	Alpub + ICDAR	randomcrop224, invert, gaussianblur, gray	76.90%	76.59%

TABLE 4.9: **Results on ResNet-50 with pretraining on Alpub dataset (with top-4 selected using strategy 1)** We report the best found augmentation and their corresponding validation and test set accuracies. We observe the baseline model achieves the best results than other two methods.

identified the top four augmentations, which yielded p-values of 0.0080, 0.01091, 0.00872, and 0.00874, respectively. Top-4 augmentations are shown below.

- (a) randomcrop198,morpho\_dilation,hflip
- (b) randomcrop198,colorjitter,hflip,invert
- (c) randomcrop198,hflip,gray
- (d) randomcrop198,invert,gaussianblur,gray

2. **Strategy 2: Best average validation accuracy.** In this strategy, we selected the top four augmentations based on the average performance across three runs. We sorted the augmentations and selected the top four from this list to compare results. The top-four augmentations from the sorted list are:

- (a) `randomcrop224,morpho_dilation,affine,colorjitter`
- (b) `randomcrop224,morpho_dilation,affine`
- (c) `randomcrop224,affine,gaussianblur`
- (d) `randomcrop224,affine,colorjitter`

The selected top four augmentations were then used to pretrain the models on the ALPUB dataset. Following this pretraining, all models were finetuned on the ICDAR dataset by adding a classification layer to the embedding models and applying the base augmentation.

**Our Findings.** We present the results of ICDAR letter recognition, leveraging pre-training on the ALPUB dataset and finetuning on ICDAR, in Tables 4.8 and 4.9 for ResNet-18 and ResNet-50, respectively. Consistent with our findings in Section A, the cross-entropy baseline achieved performances of 79.94% on ResNet-18 and 79.75% on ResNet-50, continuing to outperform the embedding-based methods. Specifically, the Triplet model attained performances of 77.51% on ResNet-18 and 77.03% on ResNet-50, while the SimCLR model reached 76.14% on ResNet-18 and 76.59% on ResNet-50. These results underscore the robustness of the cross-entropy approach, which consistently yields higher accuracy compared to the other methods tested.

Moreover, we observed that each method responded differently to the augmentations applied, with no single augmentation strategy emerging as universally optimal across all models. This variability highlights the importance of carefully selecting augmentation strategies tailored to each specific method and architecture. The impact of data augmentation on the training process is therefore significant, as it directly influences the performance outcomes for each model.

However, it is noteworthy that the final performance after finetuning on the ICDAR dataset, despite pretraining on the ALPUB dataset, reached only 79.75%, which did not surpass the 80.47% achieved in earlier experiments conducted without pretraining on the ALPUB dataset. This result is unexpected, as pretraining on a large-scale dataset like ALPUB was anticipated to improve the model’s performance on the downstream ICDAR task.

The above results are with top-4 selected with strategy 1. Furthermore, in the following, we present the results with stragey 2. Table 4.10 displays the results for the three models. The observed pattern is consistent with the pretraining results. The baseline model achieved a test accuracy of 81.14%, while the embedding-based models, such as Triplet, reached 78.88% and SimCLR achieved 79.18%. For result 50 also follows same pattern as before, In table 4.11 we can observe that baseline model is performing best than other 2 embedding models. Test accuracies and baseline model, triple model and SimCLR models are 81.17%, 78.24%, 78.85%

This unexpected result—where pretraining on the large-scale ALPUB dataset did not enhance performance on the finetuned ICDAR dataset—warrants further investigation. To better understand this outcome, we provide embedding visualizations generated from different methods to support and explain these quantitative results. These visualizations will offer deeper insights into how the representations learned during pretraining might have affected the final model performance and why the expected improvements did not materialize.

Experiment	Dataset	Best augmentation	Valid Acc.	Test Acc.
Baseline model	Alpub + ICDAR	<code>randomcrop224, morpho_dilation, affine</code>	81.62%	81.08%
Triplet model	Alpub + ICDAR	<code>randomcrop224, morpho_dilation, affine, colorjitter</code>	79.57%	78.88%
SimCLR model	Alpub + ICDAR	<code>randomcrop224, affine, gaussianblur</code>	79.55%	78.81%

TABLE 4.10: **Results on ResNet-18 with pretraining on Alpub dataset (with top-4 selected using strategy 2).** We report the best found augmentation and their corresponding validation and test set accuracies. We observe the baseline model achieves the best results than other two methods.

Experiment	Dataset	Best augmentation	Valid Acc.	Test Acc.
Baseline model	Alpub + ICDAR	<code>randomcrop224, affine, gaussianblur</code>	80.82%	<b>81.00%</b>
Triplet model	Alpub + ICDAR	<code>randomcrop224, morpho_dilation, affine, colorjitter</code>	79.17%	78.24%
SimCLR model	Alpub + ICDAR	<code>randomcrop224, morpho_dilation, affine, colorjitter</code>	78.53%	77.83%

TABLE 4.11: **Results on ResNet-50 with pretraining on Alpub dataset (with top-4 selected using strategy 2).** We report the best found augmentation and their corresponding validation and test set accuracies. We observe the baseline model achieves the best results than other two methods.

### 4.3 t-SNE Analysis

To gain a deeper understanding of the different methods, we visualize t-SNE plots of the embeddings 1. at the end of pretraining with ALPUB and 2. after finetuning with ICDAR. Specifically, we select approximately 1,000 samples from the ICDAR test set for this visualization. These embeddings are analyzed at these two stages of the training pipeline to provide insights into how the models’ representations evolve at the end of pretraining and after final finetuning.

1. **t-SNE with Baseline Model.** Figure 4.1a shows the t-SNE visualization of 1,000 ICDAR test samples from the model obtained at the end of pretraining

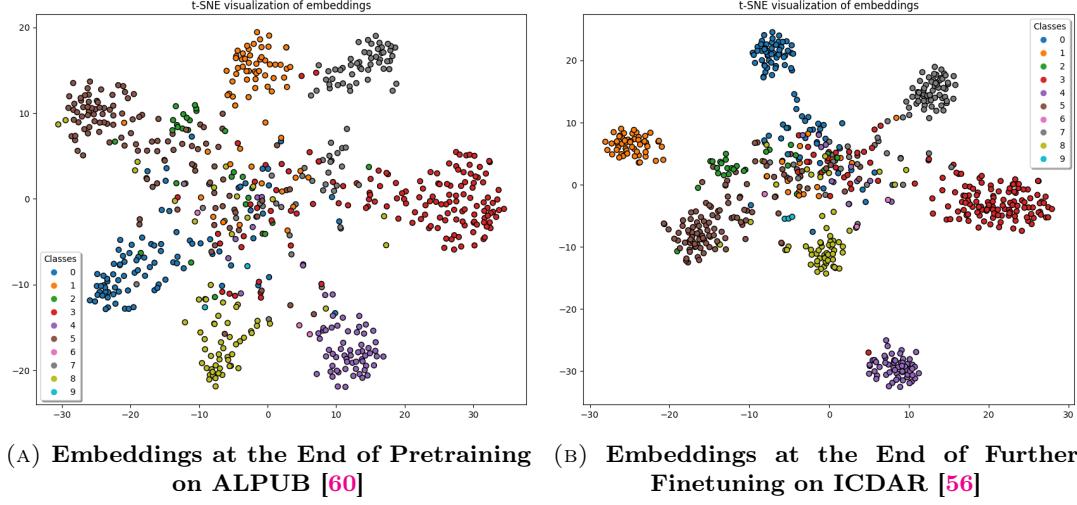
with ALPUB. Additionally, Figure 4.1b presents the embeddings after the model has been further finetuned with the ICDAR dataset. The t-SNE plots for the baseline model trained with cross-entropy loss exhibit well-defined clusters for each class both at the end of pretraining and after finetuning. This clear cluster separation indicates that the baseline model is highly effective at distinguishing between different letter classes in the embedding space, which likely contributes to its superior performance relative to the other methods.

2. **t-SNE with Triplet Model.** Figure 4.2a presents the t-SNE visualization of ICDAR test samples using the Triplet embedding model, which was pretrained on the ALPUB dataset. We observe that while the classes form distinct clusters, these clusters are not as clearly separated as those produced by the cross-entropy baseline method. This suggests that the Triplet model, though effective, may not be as precise in distinguishing between certain letter classes in the embedding space. Additionally, Figure 4.2b displays the t-SNE embeddings after the model has been finetuned with the ICDAR dataset, indicating that the cluster separations remain less distinct compared to the cross-entropy method.
3. **t-SNE with SimCLR.** Figure 4.3a shows the t-SNE visualization of ICDAR test samples using the SimCLR embedding model pretrained on the ALPUB dataset. We observe that the classes are not well-clustered, raising concerns about the convergence of the SimCLR method on the ALPUB dataset. This lack of clear clustering suggests that the SimCLR model may struggle to learn distinct class separations during the pretraining phase. In contrast, Figure 4.3b visualizes the t-SNE embeddings after finetuning with the ICDAR dataset with an additional classification layer. Here, we see that the embeddings are more clearly separated, indicating some improvement in class distinction following finetuning. However, the additional benefit of pretraining with SimCLR on the ALPUB dataset appears limited, as the downstream performance on ICDAR does not show significant enhancement. These results are consistent with the quantitative observations discussed in Section 4.2.1, where the SimCLR method underperformed relative to the baseline CE method. Overall, the lack of clear clustering patterns in the t-SNE plots after pretraining suggests that SimCLR struggles to achieve strong class separation, even with the advantage of a larger pretraining dataset.

## 4.4 Summary

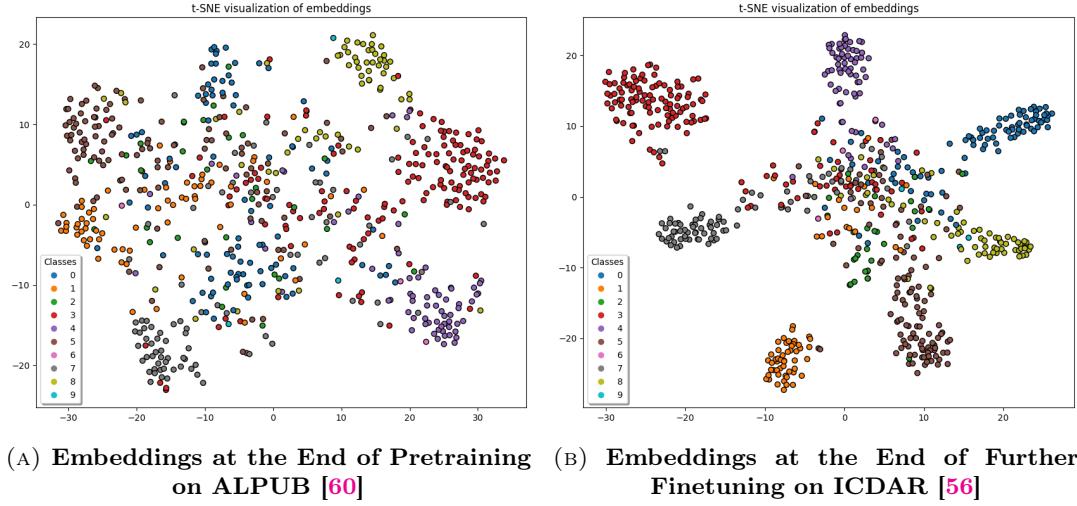
In this chapter, we conducted an extensive evaluation of various models and training strategies for Greek letter recognition using the ICDAR dataset. The experiments were designed to explore the impact of different pretraining and finetuning techniques, particularly focusing on the effectiveness of pretraining on the large-scale ALPUB dataset.

In Section 4.2, we began by examining the results of direct training on the ICDAR dataset without pretraining on the Alpub, comparing the performance of three distinct methods: the Baseline model, Triplet model [31], and SimCLR model [6]. As shown in Tables 4.6, 4.7, 4.8, 4.9 and 4.10, the Baseline model trained with cross-entropy loss consistently outperformed the embedding-based methods across different architectures, highlighting the robustness of this approach in the absence of pretraining.



(A) Embeddings at the End of Pretraining on ALPUB [60] (B) Embeddings at the End of Further Finetuning on ICDAR [56]

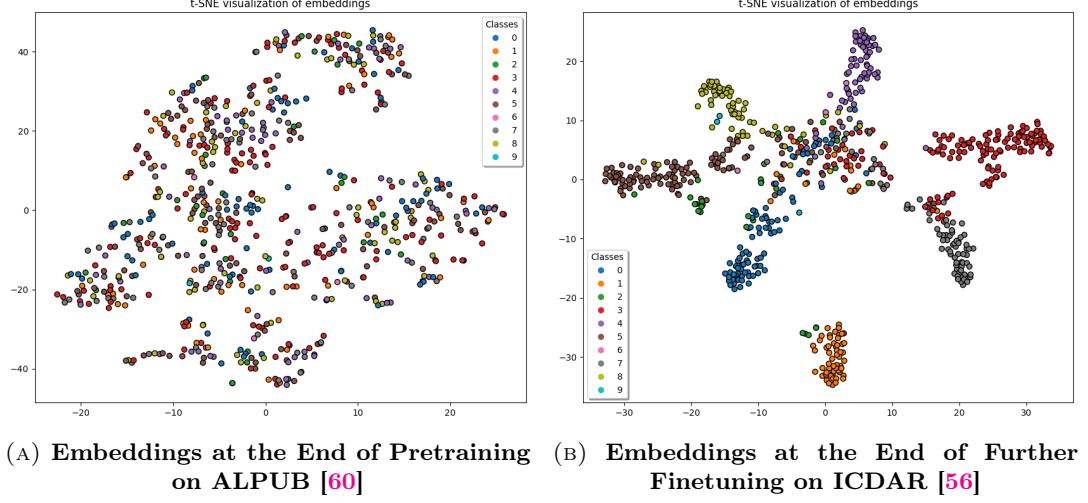
FIGURE 4.1: Comparison of t-SNE Visualizations of the Baseline Model at the end of pretraining with ALPUB (left) and after further finetuning with ICDAR (right). We visualize the embeddings of 1,000 data points from the ICDAR test set using the ResNet-18 backbone. The embeddings are derived from the feature representation just before the classification layer.



(A) Embeddings at the End of Pretraining on ALPUB [60] (B) Embeddings at the End of Further Finetuning on ICDAR [56]

FIGURE 4.2: Comparison of t-SNE Visualizations of the Triplet Model at the end of pretraining with ALPUB (left) and after further finetuning with ICDAR (right). The visualizations depict embeddings of 1,000 data points from the ICDAR test set using the ResNet-18 backbone. The pretraining and finetuning was conducted with the augmentations: `randomcrop224`, `invert`, `gaussianblur`, `gray`.

Subsequently in Section 4.1.1, we introduced pretraining on the ALPUB dataset [60], followed by finetuning on the ICDAR dataset [56]. Despite the expectation that pretraining on a larger and more diverse dataset would enhance performance, the results revealed no substantial improvements, with the cross-entropy Baseline model still achieving the best overall accuracy. This finding was further supported by t-SNE visualizations, which showed clearer class separations in the Baseline model compared to the Triplet and SimCLR models.



**FIGURE 4.3: Comparison of t-SNE Visualizations of the SimCLR Model** at the end of pretraining with ALPUB (left) and after further finetuning with ICDAR (right). The visualizations depict embeddings of 1,000 data points from the ICDAR test set using the ResNet-18 backbone. The pretraining was conducted with the augmentations: `randomcrop224`, `hflip`, `gray`.

Furthermore, in Section 4.3, the t-SNE analysis provided deeper insights into how each model’s embeddings evolved during the training process. Notably, the SimCLR model exhibited less distinct clustering of classes even after pretraining on a large scale Alpub dataset [60], suggesting that the benefits of unsupervised contrastive learning might be limited in this specific application.

Overall, this chapter underscores the importance of careful method selection and the potential limitations of certain training strategies, such as contrastive learning, in specialized tasks like Greek letter recognition. The results suggest that traditional supervised learning approaches, particularly those utilizing cross-entropy loss, may still offer the most reliable performance in such contexts. In the following chapter, we delve into potential reasons for the unexpected performance observed in our experiments and highlight the key limitations within our experimental setup.

## Chapter 5

# Discussion and Limitations

In this chapter, we will try to understand better into the results presented in the previous chapters, offering few potential reasons of the outcomes observed across the different models and training strategies. By closely examining the performance metrics and visualizations, we aim to uncover the underlying factors that contributed to the models' successes and challenges. This analysis will help clarify the broader implications of our findings and provide insights into the effectiveness of various approaches in the context of Greek letter recognition. Additionally, we will explore the key limitations inherent in our experimental setup, which may have influenced the results. Understanding these constraints is crucial for assessing the validity and generalizability of our conclusions.



**FIGURE 5.1: SimCLR cropping scheme leads to semantic shift in the labels.** For example, we observe the two views of the image cropped from the original image with 60% area. It can be seen that this cropping scheme leads to a change in the labels.

### 5.1 Discussion

**Q1. What is the impact of the cropping scheme in SimCLR?** In Figure 5.1, we observe that a spatial cropping scheme with a coverage of 60% sometimes significantly alters the semantic content of the image, often resulting in a shift from one label to another. This raises a critical question: can the standard cropping techniques commonly used in natural image recognition tasks be directly applied to the domain of letter recognition, particularly for Greek letters? Our experimental results strongly

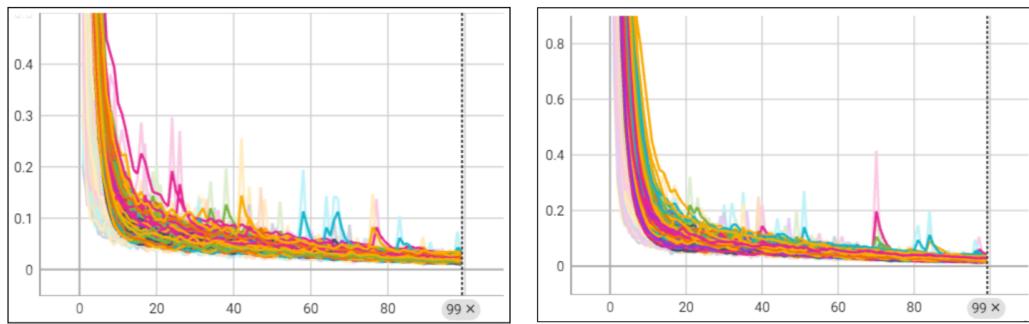


FIGURE 5.2: **SimCLR Validation loss.** Comparison between ResNet-18 (left) and ResNet-50 (right) over 20 epochs.

suggest that SimCLR underperforms relative to baseline methods, a shortcoming that can largely be attributed to the semantic shift caused by the cropping process. Conversely, when the cropped region is too large, the resulting positive pairs in SimCLR become nearly indistinguishable, causing the network to struggle in learning meaningful features. These findings indicate that current cropping techniques may not be adequate for maintaining semantic integrity in letter recognition tasks. Therefore, we argue that further research is needed to develop a cropping scheme that preserves semantic consistency, enabling the effective application of SimCLR to Greek letter recognition.

**Q2. How to verify that SimCLR training is converging?** To determine whether SimCLR training is converging effectively, we analyze the validation loss across various augmentations over multiple epochs, as shown in Figure 5.2. The decreasing trend in SimCLR loss throughout the training process suggests that the model is indeed improving. However, it is noteworthy that the pretraining of SimCLR on the extensive ALPUB dataset does not result in enhanced performance on the downstream tasks. This discrepancy may be attributed to errors introduced during the cropping phase in the pretraining stage, which could have negatively impacted the model’s ability to generalize effectively to new datasets.

## 5.2 Limitations

This thesis, while presenting evidence on the performance of three widely used methods, acknowledges several limitations inherent in the conducted research. These limitations are detailed below:

1. **Hyperparameter Tuning.** Although we made efforts to optimize the data-augmentation strategies, a number of model-specific hyperparameters, such as dropout rates and optimizer-specific parameters including learning rate, momentum, beta, and decay rate, were not exhaustively tuned due to constraints in hardware resources. Additionally, the temperature value also requires tuning. This limitation may have impacted the overall performance and generalizability of the models.

2. **Data Augmentation.** Our study explored 10 data augmentation strategies in Section 3.1; however, the Albumentations library [5] supports up to 40 augmentations, many of which were not examined in this research. Additionally, several of the augmentations implemented required the manual setting of hyperparameters, which were fixed at predefined values. A comprehensive sweep of these hyperparameters might have provided more extensive insights into the model’s performance under varied conditions.
3. **Cropping Size for SimCLR.** The decision to crop 60% of the original image area in SimCLR was heuristic and lacked a theoretical foundation. We recognize that this arbitrary choice in spatial cropping may have unintentionally altered the semantic content of the images, possibly resulting in label inconsistencies and affecting the model’s performance.
4. **Batch Size in SimCLR.** SimCLR, a popular approach in metric learning, is known for its sensitivity to batch size. Typically, during pretraining, SimCLR is applied with a substantial batch size—often around 2048. This large batch size is critical as it allows the model to effectively leverage hard negatives within the batch, which are pivotal for the model’s ability to learn meaningful representations in a self-supervised manner. However, due to computational limitations in the present experiments, the batch size was significantly reduced to 115. This reduction implies that the model may not have encountered a sufficient number of hard negatives during training, potentially impacting its ability to learn robust representations.
5. **Dataset construction** We randomly split the available dataset in three partitions: 70%-15%-15% for training, validation, and testing sets, and use the same split for all experiments. However, this potentially leads to bias and it would be worth looking into splitting multiple times and average the results over multiple runs.

## Chapter 6

# Conclusion & Future Work

### 6.1 Conclusion

This thesis evaluates the performance of SimCLR [6], a contrastive learning technique, for the task of Greek letter recognition [61] and compares it against traditional models that utilize cross-entropy and triplet loss functions [31]. A significant portion of this work (in Section 4.2) involves ablating the data augmentation strategy across a pool of 93 different augmentations, comprising both spatial and pixel-level techniques 4.1.4. Through a comprehensive analysis involving both a large pretraining dataset (ALPUB [60]) and a smaller fine-tuning competition dataset [56], it has become evident that SimCLR, despite its growing popularity in various image recognition tasks, does not outperform traditional supervised learning approaches such as cross-entropy baseline and triplet learning within the specific context of Greek letter recognition.

The unexpected underperformance of SimCLR raises intriguing questions. While it is challenging to pinpoint the exact reasons behind this result, our analysis in Chapter 4 and 5 suggests that the method of cropping sub-images to generate positive pairs in SimCLR may introduce significant semantic shifts, which can be particularly detrimental in the context of letter images. This semantic drift likely contributes to the unsatisfactory performance observed. In conclusion, our results indicate that the baseline model trained with cross-entropy loss consistently outperforms both SimCLR and the triplet loss model.

### 6.2 Future Work

In this section, we present the few avenues for future research.

- **Exploration of Additional Contrastive Learning Strategies:** Future work could investigate other contrastive learning frameworks and strategies, such as MoCo [24] or BYOL [20], to determine if they offer better performance in specialized tasks like letter recognition.
- **Hyperparameter Tuning:** Experiments with different hyperparameter values, including learning rates, batch sizes, and the temperature parameter in SimCLR’s contrastive loss function, could yield further insights into optimizing contrastive learning models for letter recognition.
- **Data Augmentation Strategies:** Additional research could focus on the impact of different data augmentation techniques, including varying cropping sizes and types of augmentations, on the performance of SimCLR and other contrastive learning models.

- **Integration with Supervised Learning:** Investigating hybrid approaches that combine contrastive learning with traditional supervised methods might provide a more robust solution, especially in cases where labeled data is limited.
- **Domain-Specific Contrastive Learning:** Future research could aim to develop contrastive learning techniques specifically tailored to the unique characteristics of Greek letters or other non-Latin scripts, potentially improving recognition accuracy in these domains.

We hope that our comprehensive empirical analysis of contrastive learning in Greek letter recognition will inspire future research aimed at developing domain-specific contrastive learning techniques tailored to such specialized tasks.

**Note:** For writing this thesis, I have taken assistance of GPT for refining the text at few places.

# Bibliography

- [1] Omar Alonso. Challenges with label quality for supervised learning. *Journal of Data and Information Quality (JDIQ)*, 6(1):1–3, 2015.
- [2] Yan Bai, Yihang Lou, Feng Gao, Shiqi Wang, Yuwei Wu, and Ling-Yu Duan. Group-sensitive triplet embedding for vehicle reidentification. *IEEE Transactions on Multimedia*, 20(9):2385–2399, 2018.
- [3] Horace B Barlow. Unsupervised learning. *Neural computation*, 1(3):295–311, 1989.
- [4] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Self-restrained triplet loss for accurate masked face recognition. *Pattern Recognition*, 124:108473, 2022.
- [5] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.
- [8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019.
- [9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.
- [10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 886–893. Ieee, 2005.
- [11] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):1–60, 2008.
- [12] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134:19–67, 2005.

- [13] Happiness Ugochi Dike, Yimin Zhou, Kranthi Kumar Deveerasetty, and Qingtian Wu. Unsupervised learning based on artificial neural network: A review. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pages 322–327. IEEE, 2018.
- [14] Antonio D’Innocente, Nikhil Garg, Yuan Zhang, Loris Bazzani, and Michael Donoser. Localized triplet loss for fine-grained fashion image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3910–3915, 2021.
- [15] Antonio D’Innocente, Nikhil Garg, Yuan Zhang, Loris Bazzani, and Michael Donoser. Localized triplet loss for fine-grained fashion image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3910–3915, 2021.
- [16] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.
- [17] Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889, 2004.
- [18] Zach Eaton-Rosen, Felix Bragman, Sebastien Ourselin, and M Jorge Cardoso. Improving data augmentation for medical image segmentation. 2018.
- [19] Zoubin Ghahramani. Unsupervised learning. In *Summer school on machine learning*, pages 72–112. Springer, 2003.
- [20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [21] Kalanit Grill-Spector and Nancy Kanwisher. Visual recognition: As soon as you know it is there, you know what it is. *Psychological Science*, 16(2):152–160, 2005.
- [22] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. *The elements of statistical learning: Data mining, inference, and prediction*, pages 485–585, 2009.
- [23] Taihei Hayashi, Keiji Gyohten, Hidehiro Ohki, and Toshiya Takami. A study of data augmentation for handwritten character recognition using deep learning. In *2018 16th International conference on frontiers in handwriting recognition (ICFHR)*, pages 552–557. IEEE, 2018.
- [24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [28] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron Van den Oord, Oriol Vinyals, and Joao Carreira. Efficient visual pretraining with contrastive detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10086–10096, 2021.
- [29] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [30] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [31] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pages 84–92. Springer, 2015.
- [32] Minui Hong, Jinwoo Choi, and Gunhee Kim. Stylemix: Separating content and style for enhanced data augmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14862–14870, 2021.
- [33] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16291–16301, 2021.
- [34] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [35] Kaer Huang, Kanokphan Lertniphonphan, Feng Chen, Jian Li, and Zhepeng Wang. Multi-object tracking by self-supervised learning appearance model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3163–3169, 2023.
- [36] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.
- [37] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.
- [38] Sangwon Kim, Jimi Lee, and Byoung Chul Ko. Ssl-mot: self-supervised learning based multi-object tracking. *Applied Intelligence*, 53(1):930–940, 2023.
- [39] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [40] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).
- [41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [42] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [43] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.
- [44] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv preprint arXiv:2011.13677*, 2020.
- [45] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [46] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.
- [47] K Markou, L Tsochatzidis, Konstantinos Zagoris, Alexandros Papazoglou, Xenofon Karagiannis, Symeon Symeonidis, and Ioannis Pratikakis. A convolutional recurrent neural network for the handwritten text recognition of historical greek manuscripts. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10-15, 2021, Proceedings, Part VII*, pages 249–262. Springer, 2021.
- [48] Ofer Matan, RK Kiang, CE Stenard, Bernhard Boser, JS Denker, Don Henderson, RE Howard, W Hubbard, LD Jackel, and Yann Le Cun. Handwritten character recognition using neural network architectures. In *4th USPS advanced technology conference*, volume 2, pages 1003–1011, 1990.
- [49] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.
- [50] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1369–1378, 2021.
- [51] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 319–345. Springer, 2020.
- [52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

- [53] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- [54] Eddy Sánchez-DelaCruz and Cecilia-Irene Loeza-Mejía. Importance and challenges of handwriting recognition with the implementation of machine learning techniques: a survey. *Applied Intelligence*, pages 1–22, 2024.
- [55] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
- [56] Mathias Seuret, Isabelle Marthot-Santaniello, Stephen A White, Olga Serbaeva Saraogi, Selaudin Agolli, Guillaume Carrière, Dalia Rodriguez-Salas, and Vincent Christlein. Icdar 2023 competition on detection and recognition of greek letters on papyri. In *International Conference on Document Analysis and Recognition*, pages 498–507. Springer, 2023.
- [57] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.
- [58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [59] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. Mixup-transformer: Dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*, 2020.
- [60] Matthew I. Swindall, Gregory Croisdale, Chase C. Hunter, Ben Keener, Alex C. Williams, James H. Brusuelas, Nita Krevans, Melissa Sellew, Lucy Fortson, and John F. Wallin. Exploring learning approaches for ancient greekcharacter recognition with citizen science data. In *2021 17th International Conference on eScience (eScience)*, pages 128–137. IEEE.
- [61] VV Satyanarayana Tallapragada, N Alivelu Manga, MV Nagabhushanam, and M Venkata naresh. Greek handwritten character recognition using inception v3. In *Smart Systems: Innovations in Computing: Proceedings of SSIC 2021*, pages 247–257. Springer, 2022.
- [62] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.
- [63] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [64] Prem Chand Vashist, Anmol Pandey, and Ashish Tripathi. A comparative study of handwriting recognition techniques. In *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, pages 456–461. IEEE, 2020.
- [65] Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. *arXiv preprint arXiv:2003.13048*, 2020.

- [66] Jingyao Wang, Luntian Mou, Changwen Zheng, and Wen Gao. Cssl-rha: Contrastive self-supervised learning for robust handwriting authentication. *arXiv preprint arXiv:2307.11100*, 2023.
- [67] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. In *Computer Vision-ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part I 6*, pages 18–32. Springer, 2000.
- [68] Enze Xie, Jian Ding, Wenhui Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhen-guo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8392–8401, 2021.
- [69] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9:611–629, 2018.
- [70] Fei Yin, Qiu-Feng Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Icdar 2013 chinese handwriting recognition competition. In *2013 12th international conference on document analysis and recognition*, pages 1464–1470. IEEE, 2013.
- [71] Xiaoyi Zhang, Tianwei Wang, Jiapeng Wang, Lianwen Jin, Canjie Luo, and Yang Xue. Chaco: character contrastive learning for handwritten text recognition. In *International Conference on Frontiers in Handwriting Recognition*, pages 345–359. Springer, 2022.
- [72] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.
- [73] Yuanyi Zhong, Bodhi Yuan, Hong Wu, Zhiqiang Yuan, Jian Peng, and Yu-Xiong Wang. Pixel contrastive-consistent semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7273–7282, 2021.
- [74] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.
- [75] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 566–583. Springer, 2020.
- [76] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 566–583. Springer, 2020.

## Appendix A

# Full Results of Experiments

In this chapter, we present all the computed results from our experiments, organized by model and training strategy.

### A.1 Baseline Model Results

#### A.1.1 ResNet-18 Architecture

Tables A.1 provides the baseline model results using the ResNet-18 architecture on the fine-tuning dataset. The best-performing augmentations were `randomcrop224`, `morpho_erosion`, `morpho_dilation`, and `gaussianblur`, achieving a validation accuracy of 81.19% and a test accuracy of 80.67%.

#### A.1.2 ResNet-50 Architecture

Tables A.2 provides the baseline model results using the ResNet-50 architecture on the fine-tuning dataset. The top augmentations were `randomcrop224`, `morpho_erosion`, and `gaussianblur`, resulting in a validation accuracy of 80.70% and a test accuracy of 80.47%.

#### A.1.3 ALPUB Dataset Evaluation

We evaluated the top four performing baseline models on the ALPUB dataset using both ResNet-18 and ResNet-50. The results are presented in Tables A.3, A.4, A.5, and A.6. For ResNet-18, the optimal augmentation combination was `randomcrop224`, `invert`, `gaussianblur`, and `gray`, yielding validation and test accuracies of 86.34% and 86.72%, respectively. Similarly, for ResNet-50, the best augmentation set was the same, resulting in validation and test accuracies of 86.47% and 86.89%, respectively. When fine-tuning on ICDAR, ResNet-18 achieved a maximum validation accuracy of 80.49% and a test accuracy of 79.94%. ResNet-50 reached a maximum validation accuracy of 80.15% and a test accuracy of 79.98%.

### A.2 Triplet Model Results

#### A.2.1 ResNet-18 Architecture

The pretraining results for the Triplet model using the ResNet-18 architecture are shown in Table A.7. Fine-tuning results on the smaller dataset, both with and without the backbone, are available in Tables A.9 and A.8.

### A.2.2 ResNet-50 Architecture

For the ResNet-50 model, pretraining on the small dataset results are shown in Table A.13. Fine-tuning results on the small dataset, with and without the backbone, are presented in Tables A.15 and A.14.

### A.2.3 ALPUB Dataset Evaluation

The results of the Triplet model on the ALPUB dataset are provided in Tables A.10, A.12, A.11, A.16, A.18, and A.17.

## A.3 SimCLR Model Results

### A.3.1 ResNet-18 Architecture

The pretraining results for the SimCLR model using the ResNet-18 architecture are shown in Table A.19. Fine-tuning results on the smaller dataset, both with and without the backbone, are available in Tables A.21 and A.20.

### A.3.2 ResNet-50 Architecture

For the ResNet-50 model, pretraining on the ALPUB dataset results are shown in Table A.25. Fine-tuning results on the small dataset, with and without the backbone, are presented in Tables A.27 and A.26.

### A.3.3 ALPUB Dataset Evaluation

SimCLR pretraining on the ALPUB dataset results for both ResNet-18 and ResNet-50 are available in Tables A.22 and A.28. Fine-tuning on the small dataset results for both ResNet-18 and ResNet-50, with and without backbone, are available in Tables A.24, A.23, A.30, and A.29.

## A.4 Baseline Results with Cross-Entropy Loss Using ResNet-18 and ResNet-50 Networks

### A.4.1 Baseline using ResNet-18 architecture on ICDAR Dataset

No	Transform Type	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho\_erosion,invert	81.81%	79.54%	79.63%
2	randomcrop224,morpho\_erosion,colorjitter,gray	83.70%	79.74%	79.00%
3	randomcrop224,morpho\_erosion,gaussianblur	83.62%	80.39%	80.27%
4	randomcrop224,morpho\_erosion,affine,gaussianblur	81.15%	80.66%	79.71%
5	randomcrop224,affine,colorjitter,gray	79.35%	79.94%	79.02%
6	randomcrop224,affine,hflip,gray	79.46%	79.29%	78.81%
7	randomcrop224,affine,gray	78.89%	79.92%	79.59%
8	randomcrop224,invert,gaussianblur,gray	79.95%	78.96%	79.30%
9	randomcrop224,morpho\_erosion,affine,gray	83.37%	79.68%	79.28%
10	randomcrop224,morpho\_erosion,colorjitter,gaussianblur	82.67%	80.37%	80.00%
11	randomcrop224,morpho\_dilation,hflip,gray	83.01%	79.17%	78.87%
12	randomcrop224,affine,hflip,invert	78.78%	78.86%	78.41%

No.	Augmentations	Train Acc	Validation Acc	Test Acc
13	randomcrop224,hflip,gray	82.15%	79.41%	78.57%
14	randomcrop224,hflip,invert	82.03%	79.51%	78.92%
15	randomcrop224,morpho_erosion,invert,gaussianblur	85.37%	79.84%	79.59%
16	randomcrop224,affine,hflip,gaussianblur	82.02%	79.78%	79.32%
17	randomcrop224,affine,colorjitter,invert	82.55%	79.57%	79.39%
18	randomcrop224,morpho_dilation,affine,hflip	81.30%	79.02%	78.71%
19	randomcrop224,morpho_erosion,affine,colorjitter	83.67%	80.09%	79.63%
20	randomcrop224,affine,colorjitter	83.42%	80.35%	79.47%
21	randomcrop224,invert,gray	80.08%	78.59%	78.40%
22	randomcrop224,morpho_erosion,morpho_dilation,hflip	82.38%	79.17%	78.51%
23	randomcrop224,colorjitter,invert,gray	78.46%	78.53%	78.63%
24	randomcrop224,morpho_dilation,gaussianblur	82.86%	79.98%	79.39%
25	randomcrop224,morpho_dilation,affine,gray	81.13%	80.07%	80.12%
26	randomcrop224,hflip,gaussianblur	83.06%	78.86%	78.36%
27	randomcrop224,morpho_erosion,morpho_dilation,invert	81.65%	79.86%	79.61%
28	randomcrop224,invert,gaussianblur	83.57%	79.19%	78.45%
29	randomcrop224,morpho_erosion,colorjitter,hflip	80.18%	79.15%	78.45%
30	randomcrop224,morpho_dilation,affine,invert	86.17%	79.92%	79.75%
31	randomcrop224,invert	83.72%	79.82%	79.57%
32	randomcrop224,morpho_dilation,invert,gray	83.90%	78.86%	78.71%
33	randomcrop224,colorjitter,gaussianblur,gray	84.97%	79.10%	79.32%
34	randomcrop224,morpho_erosion,hflip,invert	81.62%	79.45%	78.20%
35	randomcrop224,morpho_dilation,affine,colorjitter	83.52%	80.27%	80.02%
36	randomcrop224,colorjitter,hflip,gaussianblur	80.20%	79.08%	79.00%
37	randomcrop224,morpho_dilation,invert	87.55%	79.31%	79.47%
38	randomcrop224,morpho_erosion,gray	86.24%	79.53%	79.57%
39	randomcrop224,morpho_dilation,affine,gaussianblur	84.59%	80.62%	80.22%
40	randomcrop224,gaussianblur,gray	86.89%	79.53%	79.06%
41	randomcrop224,morpho_dilation,gaussianblur,gray	82.41%	80.17%	80.12%
42	randomcrop224,morpho_dilation,colorjitter	82.09%	79.70%	79.37%
43	randomcrop224,affine,gaussianblur	84.67%	80.51%	80.59%
44	randomcrop224,morpho_erosion,morpho_dilation,affine	80.40%	80.35%	80.12%
45	randomcrop224,morpho_erosion,colorjitter	91.15%	79.94%	78.88%
46	randomcrop224,morpho_erosion,hflip,gaussianblur	82.58%	79.43%	78.18%
47	randomcrop224,morpho_erosion,hflip,gray	81.58%	78.68%	78.41%
48	randomcrop224,affine,colorjitter,hflip	80.00%	78.88%	78.45%
49	randomcrop224,colorjitter,hflip	81.80%	79.39%	78.40%
50	randomcrop224	84.20%	80.31%	79.92%
51	randomcrop224,affine,invert	78.98%	79.94%	79.84%
52	randomcrop224,morpho_erosion,affine,hflip	79.41%	79.70%	78.69%
53	randomcrop224,morpho_dilation,colorjitter,invert	78.97%	79.41%	78.67%
54	randomcrop224,morpho_erosion,affine,invert	83.13%	79.88%	79.45%
55	randomcrop224,colorjitter,invert	79.75%	79.47%	79.55%
56	randomcrop224,morpho_dilation,hflip,invert	83.03%	78.96%	78.00%
57	randomcrop224,gaussianblur	83.97%	80.31%	80.31%
58	randomcrop224,morpho_dilation,hflip,gaussianblur	82.30%	79.15%	78.30%
59	randomcrop224,morpho_dilation	83.61%	80.04%	79.94%
60	randomcrop224,morpho_dilation,invert,gaussianblur	82.03%	80.05%	79.69%
61	randomcrop224,morpho_dilation,colorjitter,hflip	81.64%	78.82%	78.57%

No.	Augmentations	Train Acc	Validation Acc	Test Acc
62	randomcrop224,morpho_dilation,hflip	82.61%	79.29%	78.75%
63	randomcrop224,colorjitter	83.42%	80.21%	79.94%
64	randomcrop224,affine,invert,gaussianblur	80.68%	80.02%	80.02%
65	randomcrop224,morpho_erosion	85.70%	80.47%	79.84%
66	randomcrop224,affine,gaussianblur,gray	79.33%	80.05%	79.82%
67	randomcrop224,morpho_erosion,affine	85.31%	80.21%	79.86%
68	randomcrop224,affine,colorjitter,gaussianblur	82.03%	79.92%	80.39%
69	randomcrop224,morpho_dilation,gray	82.49%	79.80%	79.73%
70	randomcrop224,morpho_erosion,morpho_dilation	83.11%	80.74%	80.12%
71	randomcrop224,colorjitter,gaussianblur	82.59%	80.45%	80.29%
72	randomcrop224,colorjitter,hflip,invert	82.73%	78.25%	77.42%
73	randomcrop224,affine,invert,gray	81.59%	79.47%	78.83%
74	randomcrop224,colorjitter,invert,gaussianblur	80.71%	79.72%	78.90%
75	randomcrop224,hflip,invert,gaussianblur	81.49%	78.41%	78.32%
76	randomcrop224,morpho_erosion,invert,gray	79.94%	78.57%	78.45%
77	randomcrop224,hflip,invert,gray	80.09%	78.51%	78.22%
78	randomcrop224,morpho_erosion,colorjitter,invert	78.51%	79.10%	78.85%
79	randomcrop224,morpho_dilation,colorjitter,gray	81.95%	79.60%	79.24%
80	randomcrop224,morpho_erosion,morpho_dilation,colorjitter	80.64%	80.11%	80.55%
81	randomcrop224,morpho_erosion,morpho_dilation,gray	85.65%	80.21%	79.67%
82	randomcrop224,morpho_dilation,colorjitter,gaussianblur	86.00%	80.17%	80.51%
83	randomcrop224,affine	85.76%	79.72%	79.80%
84	randomcrop224,morpho_erosion,hflip	80.79%	79.57%	78.87%
85	randomcrop224,gray	86.92%	79.60%	79.43%
86	randomcrop224,affine,hflip	80.30%	79.35%	79.22%
87	randomcrop224,morpho_erosion,gaussianblur,gray	84.02%	79.57%	79.49%
88	randomcrop224,hflip,gaussianblur,gray	80.22%	79.08%	78.67%
89	randomcrop224,morpho_dilation,affine	81.02%	80.66%	80.02%
90	<b>randomcrop224,morpho_erosion,morpho_dilation,gaussianblur</b>	<b>83.32%</b>	<b>81.19%</b>	<b>80.67%</b>
91	randomcrop224,colorjitter,hflip,gray	83.68%	79.70%	79.47%
92	randomcrop224,hflip	79.31%	79.27%	77.89%
93	randomcrop224,colorjitter,gray	82.90%	79.94%	79.59%

TABLE A.1: Performance of Baseline model ResNet-18 architecture with Cross-Entropy Loss Across Various Data Augmentation Combinations on the ICDAR Dataset

#### A.4.2 Baseline using ResNet-50 architecture on ICDAR Dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,affine,hflip	78.15%	79.12%	78.34%
2	randomcrop224,colorjitter,invert	85.25%	79.49%	79.14%
3	randomcrop224,morpho_erosion,morpho_dilation,colorjitter	81.12%	80.29%	80.39%
4	randomcrop224,morpho_dilation,affine,colorjitter	80.87%	80.29%	80.18%
5	randomcrop224,affine,gaussianblur,gray	80.24%	80.15%	79.61%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
6	randomcrop224,colorjitter,invert,gray	82.48%	79.51%	78.90%
7	randomcrop224,colorjitter,gray	82.66%	80.64%	80.35%
8	randomcrop224,morpho_dilation,affine,hflip	78.85%	78.27%	77.75%
9	randomcrop224,colorjitter,invert,gaussianblur	82.24%	79.59%	79.71%
10	randomcrop224,affine,invert,gaussianblur	79.95%	80.23%	80.12%
11	randomcrop224,affine,hflip,gaussianblur	78.96%	79.29%	78.75%
12	randomcrop224,morpho_erosion,morpho_dilation	83.05%	80.41%	79.92%
13	randomcrop224,morpho_erosion,invert,gray	80.50%	79.33%	78.63%
14	randomcrop224,morpho_dilation,hflip,gaussianblur	82.46%	78.94%	78.98%
15	randomcrop224,affine,gaussianblur	83.89%	80.50%	79.67%
16	randomcrop224,affine	81.35%	80.35%	80.02%
17	randomcrop224,affine,hflip,invert	76.17%	78.06%	77.53%
18	randomcrop224,morpho_erosion,morpho_dilation,affine	83.97%	80.68%	80.29%
19	randomcrop224,morpho_erosion,colorjitter,invert	83.95%	79.74%	78.90%
20	randomcrop224,colorjitter,gaussianblur,gray	82.99%	79.88%	80.10%
21	randomcrop224,affine,invert,gray	79.34%	79.45%	79.35%
22	randomcrop224,morpho_dilation,hflip,invert	80.55%	79.02%	78.92%
23	randomcrop224,morpho_dilation,gaussianblur	86.15%	80.25%	79.94%
24	randomcrop224,morpho_erosion,affine,invert	81.96%	80.07%	79.69%
25	randomcrop224,hflip	81.90%	79.13%	78.06%
26	randomcrop224,hflip,gaussianblur,gray	81.58%	79.10%	79.08%
27	randomcrop224,morpho_erosion,morpho_dilation,invert	85.68%	80.49%	79.77%
28	randomcrop224,morpho_erosion,invert,gaussianblur	83.28%	79.98%	79.75%
29	randomcrop224,invert,gaussianblur	84.89%	80.49%	79.82%
30	randomcrop224,colorjitter,gaussianblur	85.49%	80.62%	80.04%
31	randomcrop224,morpho_erosion,affine,gaussianblur	83.25%	80.37%	80.61%
32	randomcrop224,morpho_erosion,hflip,invert	81.07%	78.76%	78.45%
33	randomcrop224,affine,colorjitter	83.05%	80.05%	80.27%
34	randomcrop224,invert,gray	83.12%	79.74%	79.00%
35	randomcrop224,morpho_dilation,affine,gray	81.02%	79.88%	79.57%
36	randomcrop224,morpho_dilation	80.29%	80.25%	79.45%
37	randomcrop224,morpho_erosion,gray	82.38%	80.05%	79.73%
38	randomcrop224,morpho_dilation,colorjitter,hflip	79.56%	78.57%	78.51%
39	randomcrop224,morpho_dilation,colorjitter,gaussianblur	84.78%	80.35%	79.84%
40	randomcrop224,colorjitter,hflip,gaussianblur	80.40%	79.12%	77.81%
41	randomcrop224,morpho_erosion,affine,gray	81.84%	80.25%	79.75%
42	randomcrop224,colorjitter,hflip	81.14%	79.27%	78.63%
43	randomcrop224,hflip,invert	81.24%	78.67%	77.79%
44	randomcrop224,morpho_dilation,affine,gaussianblur	82.35%	80.52%	80.43%
45	randomcrop224,morpho_dilation,invert,gray	82.98%	80.25%	79.30%
46	randomcrop224,affine,hflip,gray	78.56%	79.23%	78.49%
47	randomcrop224,gray	83.78%	80.17%	80.08%
48	randomcrop224,morpho_erosion,invert	81.73%	79.62%	80.14%
49	randomcrop224,morpho_dilation,invert,gaussianblur	82.63%	80.33%	80.12%
50	randomcrop224,colorjitter,hflip,invert	80.20%	78.39%	78.22%
51	randomcrop224,morpho_erosion,hflip	80.81%	78.67%	78.16%
52	randomcrop224,invert	83.34%	79.92%	79.71%
53	randomcrop224,morpho_dilation,colorjitter	85.60%	80.17%	80.74%
54	randomcrop224,morpho_erosion,colorjitter,gray	82.43%	80.05%	80.43%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
55	randomcrop224,colorjitter,hflip,gray	80.79%	78.47%	78.16%
56	randomcrop224,colorjitter	84.27%	80.47%	80.08%
57	randomcrop224,hflip,gray	81.62%	78.25%	78.26%
58	randomcrop224,morpho_dilation,hflip	81.61%	79.12%	79.18%
59	randomcrop224,morpho_erosion,colorjitter	81.28%	79.84%	79.28%
60	randomcrop224,affine,invert	81.42%	80.13%	80.22%
61	randomcrop224,morpho_erosion,colorjitter,hflip	81.02%	79.04%	78.34%
62	randomcrop224,affine,colorjitter,hflip	77.53%	78.82%	78.92%
63	randomcrop224,morpho_dilation,gray	84.27%	80.02%	79.67%
64	randomcrop224,morpho_dilation,affine,invert	80.90%	80.09%	80.43%
65	randomcrop224,morpho_erosion,gaussianblur,gray	84.54%	79.98%	79.16%
66	randomcrop224,morpho_erosion,hflip,gaussianblur	80.91%	78.96%	78.38%
67	randomcrop224,morpho_erosion,affine	80.99%	79.80%	80.25%
68	randomcrop224,hflip,gaussianblur	82.71%	78.59%	78.90%
69	randomcrop224,morpho_erosion,hflip,gray	80.23%	78.67%	77.79%
70	randomcrop224,morpho_dilation,colorjitter,gray	84.72%	80.05%	79.18%
71	randomcrop224,hflip,invert,gray	79.26%	78.35%	78.51%
72	randomcrop224,invert,gaussianblur,gray	80.95%	79.19%	78.86%
73	randomcrop224,morpho_erosion,morpho_dilation,gaussianblur	86.49%	80.45%	79.37%
74	randomcrop224,morpho_dilation,colorjitter,invert	85.39%	79.49%	79.16%
75	randomcrop224,affine,colorjitter,gaussianblur	82.64%	80.50%	80.29%
76	randomcrop224,affine,colorjitter,invert	80.61%	79.47%	78.85%
77	randomcrop224,morpho_dilation,gaussianblur,gray	82.92%	80.15%	80.33%
78	randomcrop224,affine,gray	81.41%	80.21%	79.35%
79	randomcrop224,affine,colorjitter,gray	82.50%	79.88%	79.69%
80	randomcrop224,morpho_erosion,morpho_dilation,gray	82.82%	79.90%	80.16%
81	randomcrop224,morpho_dilation,invert	81.69%	80.33%	80.49%
82	randomcrop224,morpho_dilation,hflip,gray	81.23%	79.13%	78.38%
83	randomcrop224,morpho_erosion	86.13%	79.96%	80.00%
84	randomcrop224,morpho_erosion,affine,hflip	77.39%	78.23%	77.89%
85	randomcrop224,hflip,invert,gaussianblur	81.77%	78.06%	77.53%
86	randomcrop224,morpho_erosion,morpho_dilation,hflip	81.66%	78.98%	79.00%
87	randomcrop224,gaussianblur,gray	83.83%	80.04%	80.02%
88	randomcrop224,morpho_erosion,affine,colorjitter	81.63%	80.54%	80.23%
89	randomcrop224	83.55%	80.45%	80.08%
90	randomcrop224,gaussianblur	82.59%	80.13%	79.88%
91	randomcrop224,morpho_dilation,affine	83.19%	80.21%	80.43%
92	randomcrop224,morpho_erosion,colorjitter,gaussianblur	85.42%	80.19%	79.59%
<b>93</b>	<b>randomcrop224,morpho_erosion,gaussianblur</b>	<b>85.42%</b>	<b>80.70%</b>	<b>80.47%</b>

TABLE A.2: Performance of Baseline model using ResNet-50 architecture with Cross-Entropy Loss across various Data Augmentation combinations on the ICDAR Dataset

#### A.4.3 Baseline using ResNet-18 architecture pertaining on Alpub Dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,hflip,gray	86.90%	85.82%	86.02%
2	randomcrop224,colorjitter,hflip,invert	87.83%	85.70%	86.15%
3	randomcrop224,morpho_dilation,hflip	88.43%	86.02%	86.34%
<b>4</b>	<b>randomcrop224,invert,gaussianblur,gray</b>	<b>90.24%</b>	<b>86.34%</b>	<b>86.72%</b>
5	randomcrop224,morpho_dilation,affine,colorjitter	88.58%	86.96%	86.99%
6	randomcrop224, morpho_dilation, affine	88.10%	87.00%	87.00%
7	randomcrop224, affine, gaussianblur	88.28%	86.79%	87.23%
8	randomcrop224, affine, colorjitter	86.86%	86.84%	87.07%

TABLE A.3: Performance of Baseline ResNet-18 Model with Cross-Entropy Loss Across Various Data Augmentation Combinations pertaining on Alpub Dataset

#### A.4.4 Baseline using ResNet-18 architecture pertaining on Alpub Dataset & fine-tuning on ICDAR dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,colorjitter,hflip,invert	86.44%	79.62%	78.83%
<b>2</b>	<b>randomcrop224,hflip,gray</b>	<b>84.16%</b>	<b>80.49%</b>	<b>79.94%</b>
3	randomcrop224,invert,gaussianblur,gray	82.48%	79.62%	80.39%
4	randomcrop224,morpho_dilation,hflip	83.14%	80.29%	80.47%
5	randomcrop224,morpho_dilation,affine,colorjitter	80.07%	80.41%	79.67%
<b>6</b>	<b>randomcrop224, morpho_dilation, affine</b>	<b>85.09%</b>	<b>81.62%</b>	<b>81.08%</b>
7	randomcrop224, affine, gaussianblur	82.58%	80.56%	81.19%
8	randomcrop224, affine, colorjitter	85.89%	81.07%	80.53%

TABLE A.4: Performance of Baseline ResNet-18 Model with Cross-Entropy Loss Across Various Data Augmentation Combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset With backbone

#### A.4.5 Baseline using ResNet-50 architecture on Alpub Dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,hflip,gray	88.24%	85.73%	86.16%
<b>2</b>	<b>randomcrop224,invert,gaussianblur,gray</b>	<b>88.34%</b>	<b>86.47%</b>	<b>86.89%</b>
3	randomcrop224,morpho_dilation,hflip	88.05%	85.84%	85.90%
4	randomcrop224,colorjitter,hflip,invert	88.17%	85.67%	85.93%
<b>5</b>	<b>randomcrop224,morpho_dilation,affine,colorjitter</b>	<b>89.38%</b>	<b>86.98%</b>	<b>86.77%</b>
6	randomcrop224, morpho_dilation, affine	90.19%	86.89%	87.01%
7	randomcrop224, affine, gaussianblur	89.26%	86.76%	86.89%
8	randomcrop224, affine, colorjitter	89.81%	86.94%	87.23%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
-----	----------------	-----------	----------------	----------

TABLE A.5: Performance of Baseline ResNet-50 Model with Cross-Entropy Loss Across various Data Augmentation combinations pre-training Alpub Dataset

#### A.4.6 Baseline using ResNet-50 architecture on Alpub Dataset & fine-tuning on ICDAR dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	<b>randomcrop224,morpho_dilation,hflip</b>	<b>86.00%</b>	<b>80.21%</b>	<b>79.75%</b>
2	randomcrop224,invert,gaussianblur,gray	83.99%	80.15%	79.98%
3	randomcrop224,hflip,gray	80.27%	80.04%	79.65%
4	randomcrop224,colorjitter,hflip,invert	81.40%	79.78%	79.82%
5	randomcrop224,morpho_dilation,affine,colorjitter	83.20%	81.11%	80.92%
6	randomcrop224, morpho_dilation, affine	83.96%	81.56%	80.98%
7	<b>randomcrop224, affine, gaussianblur</b>	<b>80.26%</b>	<b>80.82%</b>	<b>81.00%</b>
8	randomcrop224, affine, colorjitter	80.12%	81.03%	80.82%

TABLE A.6: Performance of Baseline ResNet-50 Model with Cross-Entropy Loss Across various Data Augmentation combinations pre-training Alpub Dataset & fine-tuning on ICDAR dataset With backbone

### A.5 Baseline with Triplet embedding model Using ResNet-18 and ResNet-50 Networks

#### A.5.1 Triplet Model using ResNet-18 Pre-training on ICDAR Dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_dilation,invert,gaussianblur	51.97%	52.26%	51.44%
2	randomcrop224,morpho_dilation,invert	97.84%	82.13%	79.41%
3	randomcrop224,affine,colorjitter,gray	92.41%	70.97%	68.50%
4	randomcrop224,affine,hflip,gaussianblur	51.59%	55.78%	50.89%
5	randomcrop224,colorjitter,hflip,invert	48.09%	51.81%	47.97%
6	randomcrop224,hflip,gaussianblur	74.19%	82.68%	81.20%
7	randomcrop224,invert	98.01%	79.86%	77.13%
8	randomcrop224,affine,colorjitter	52.43%	53.06%	52.63%
9	randomcrop224,colorjitter,invert	98.64%	79.74%	77.28%
10	randomcrop224,invert,gaussianblur	50.67%	51.93%	48.96%
11	randomcrop224,morpho_dilation	72.03%	81.50%	80.11%
12	randomcrop224,morpho_erosion,colorjitter,gray	94.57%	78.14%	76.84%
13	randomcrop224,colorjitter	97.45%	81.88%	77.48%
14	randomcrop224,gray	96.93%	81.86%	78.27%
15	randomcrop224,morpho_dilation,colorjitter,invert	51.12%	53.04%	52.58%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
16	randomcrop224,morpho_dilation,affine,gaussianblur	97.92%	81.70%	76.64%
17	randomcrop224,morpho_dilation,gray	51.49%	52.36%	51.29%
18	randomcrop224,morpho_erosion,hflip,invert	97.51%	82.33%	80.31%
19	randomcrop224,invert,gaussianblur,gray	50.16%	53.30%	51.24%
20	randomcrop224,affine,gaussianblur,gray	51.45%	53.20%	48.86%
21	randomcrop224,hflip,invert,gaussianblur	49.91%	52.18%	48.31%
22	randomcrop224,morpho_erosion,affine	98.67%	81.52%	79.02%
23	randomcrop224,morpho_dilation,gaussianblur,gray	93.68%	81.25%	79.22%
24	randomcrop224,morpho_dilation,invert,gray	52.15%	52.63%	52.38%
25	randomcrop224,morpho_erosion,morpho_dilation,hflip	51.33%	53.16%	52.28%
26	randomcrop224,morpho_erosion,morpho_dilation	99.02%	81.86%	77.93%
27	randomcrop224,morpho_erosion,morpho_dilation,invert	51.73%	52.95%	50.99%
28	randomcrop224,morpho_erosion,colorjitter,invert	52.99%	52.52%	50.40%
29	randomcrop224,morpho_erosion,invert,gray	51.10%	52.91%	50.99%
30	randomcrop224,hflip,invert	49.61%	52.18%	46.78%
31	randomcrop224,morpho_dilation,affine	51.33%	53.44%	51.54%
32	randomcrop224,morpho_erosion,gaussianblur,gray	98.50%	81.93%	80.41%
33	randomcrop224,morpho_dilation,hflip	51.72%	53.24%	50.60%
34	randomcrop224,colorjitter,hflip,gray	50.57%	52.87%	51.79%
35	randomcrop224,morpho_dilation,colorjitter	98.09%	82.34%	78.67%
36	randomcrop224,morpho_dilation,hflip,gaussianblur	97.71%	83.03%	81.45%
37	randomcrop224,colorjitter,gray	98.50%	80.45%	76.09%
38	randomcrop224,invert,gray	49.75%	52.40%	49.95%
39	randomcrop224,gaussianblur	99.38%	82.85%	78.67%
40	randomcrop224,affine,colorjitter,gaussianblur	51.17%	53.14%	50.25%
41	randomcrop224,morpho_erosion,morpho_dilation,gray	98.98%	82.34%	77.93%
42	randomcrop224,morpho_erosion,morpho_dilation,affine	51.28%	53.18%	51.84%
43	randomcrop224,morpho_erosion,gaussianblur	99.56%	81.27%	77.83%
44	randomcrop224,hflip,gaussianblur,gray	97.51%	82.19%	81.10%
45	randomcrop224,morpho_erosion,morpho_dilation,colorjitter	51.04%	52.85%	54.02%
46	randomcrop224,morpho_dilation,colorjitter,gray	51.45%	53.94%	51.98%
47	randomcrop224,morpho_dilation,gaussianblur	98.87%	81.84%	77.88%
48	randomcrop224,morpho_erosion,hflip,gray	51.47%	53.30%	49.65%
49	randomcrop224,morpho_dilation,affine,colorjitter	51.37%	53.67%	50.74%
50	randomcrop224,hflip,gray	98.51%	81.11%	79.12%
51	randomcrop224,affine	90.67%	82.72%	80.41%
52	randomcrop224,morpho_dilation,colorjitter,hflip	97.43%	83.42%	80.21%
53	randomcrop224,morpho_erosion,morpho_dilation,gaussianblur	98.07%	82.76%	81.15%
54	randomcrop224,gaussianblur,gray	99.19%	82.29%	78.37%
55	randomcrop224,colorjitter,invert,gaussianblur	51.39%	51.93%	52.53%
56	randomcrop224,morpho_dilation,hflip,gray	51.18%	54.06%	50.74%
57	randomcrop224,morpho_erosion,invert,gaussianblur	99.46%	82.01%	77.28%
58	randomcrop224,morpho_erosion,affine,gray	51.47%	52.81%	48.76%
59	randomcrop224,colorjitter,hflip	95.06%	81.48%	79.76%
60	randomcrop224,affine,hflip,invert	51.80%	52.48%	48.51%
61	randomcrop224,morpho_erosion,affine,invert	51.35%	52.87%	47.32%
62	randomcrop224,morpho_dilation,affine,hflip	51.81%	54.30%	49.80%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
63	randomcrop224,morpho\_erosion,gray	95.29%	81.23%	78.62%
64	randomcrop224,colorjitter,hflip,gaussianblur	51.88%	53.06%	50.60%
65	randomcrop224,affine,invert	51.93%	51.99%	51.79%
66	randomcrop224,colorjitter,invert,gray	52.03%	51.79%	51.88%
67	randomcrop224,affine,invert,gaussianblur	51.84%	53.36%	49.95%
68	randomcrop224,morpho\_erosion,hflip	99.28%	81.82%	80.61%
69	randomcrop224,morpho\_dilation,hflip,invert	51.31%	51.99%	50.05%
70	randomcrop224,affine,gaussianblur	51.90%	53.71%	52.33%
71	randomcrop224,colorjitter,gaussianblur,gray	98.74%	81.97%	78.72%
72	randomcrop224,morpho\_erosion,invert	50.56%	52.75%	52.98%
73	randomcrop224,morpho\_erosion,colorjitter,gaussianblur	98.03%	82.48%	80.51%
74	randomcrop224,affine,hflip	97.12%	80.00%	77.93%
75	randomcrop224,hflip,invert,gray	54.89%	51.95%	48.16%
76	randomcrop224,morpho\_erosion,colorjitter,hflip	51.09%	51.61%	50.30%
77	randomcrop224,morpho\_dilation,affine,invert	51.17%	52.81%	47.72%
78	randomcrop224,affine,invert,gray	51.74%	53.16%	49.01%
79	randomcrop224,morpho\_erosion	98.92%	82.56%	78.82%
80	randomcrop224,affine,colorjitter,hflip	51.99%	53.49%	48.96%
81	randomcrop224,affine,gray	49.82%	52.65%	49.75%
82	randomcrop224,hflip	96.59%	81.72%	78.13%
83	randomcrop224,morpho\_erosion,affine,colorjitter	51.56%	53.51%	48.61%
84	randomcrop224,morpho\_dilation,affine,gray	97.65%	79.13%	76.64%
85	randomcrop224,morpho\_erosion,affine,gaussianblur	52.07%	54.67%	49.26%
86	randomcrop224	99.01%	82.68%	80.31%
87	randomcrop224,affine,colorjitter,invert	49.43%	53.06%	49.70%
88	randomcrop224,morpho\_erosion,affine,hflip	96.64%	81.31%	78.87%
89	randomcrop224,affine,hflip,gray	51.71%	52.69%	48.12%
90	randomcrop224,morpho\_erosion,colorjitter	97.17%	80.60%	78.52%
91	randomcrop224,morpho\_dilation,colorjitter,gaussianblur	51.25%	53.24%	52.08%
92	randomcrop224,morpho\_erosion,hflip,gaussianblur	98.19%	81.70%	78.32%
93	<b>randomcrop224,colorjitter,gaussianblur</b>	<b>86.47%</b>	<b>83.52%</b>	<b>81.80%</b>

TABLE A.7: Performance of Triplet ResNet-18 Model with triplet loss across various Data Augmentation combinations pertaining on ICDAR Dataset

### A.5.2 Triplet Model Using ResNet-18 fine-tuning on ICDAR Dataset Without Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho\_dilation,affine	10.92%	12.12%	11.35%
2	randomcrop224,morpho\_dilation,hflip	11.49%	12.53%	11.82%
3	randomcrop224,morpho\_dilation,affine,gray	46.75%	55.24%	53.70%
4	randomcrop224,morpho\_erosion,morpho\_dilation,colorjitter	11.28%	12.17%	11.33%
5	randomcrop224,affine,colorjitter,gray	27.51%	31.85%	31.51%
6	randomcrop224,morpho\_dilation,affine,invert	11.35%	12.31%	10.90%
7	randomcrop224,affine,hflip,invert	10.77%	12.16%	11.49%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
8	randomcrop224,colorjitter,hflip,invert	11.20%	12.16%	11.37%
9	randomcrop224,morpho_erosion,morpho_dilation,gray	55.23%	59.03%	58.98%
10	randomcrop224,morpho_erosion,affine,invert	11.35%	12.39%	11.72%
11	randomcrop224,invert,gaussianblur,gray	11.17%	12.47%	11.45%
12	randomcrop224,invert,gray	11.82%	12.37%	11.53%
13	randomcrop224,morpho_dilation,hflip,gray	11.48%	12.82%	11.78%
14	randomcrop224,morpho_erosion,morpho_dilation,invert	10.79%	12.12%	11.33%
15	randomcrop224,morpho_dilation,hflip,gaussianblur	54.42%	56.35%	57.01%
16	randomcrop224,morpho_erosion,colorjitter	56.08%	58.99%	58.90%
17	randomcrop224,morpho_dilation,invert,gray	10.97%	12.12%	11.35%
18	randomcrop224,morpho_erosion,affine,colorjitter	11.51%	12.64%	11.70%
19	randomcrop224,hflip	53.20%	54.06%	54.93%
20	randomcrop224,morpho_erosion,gaussianblur	56.44%	59.56%	59.49%
21	randomcrop224,gaussianblur,gray	56.64%	59.87%	59.67%
22	randomcrop224,morpho_dilation,colorjitter,gray	11.37%	12.37%	11.41%
23	randomcrop224,colorjitter,hflip,gray	14.93%	16.30%	15.40%
24	randomcrop224,colorjitter,gaussianblur,gray	53.30%	55.29%	54.52%
25	randomcrop224,colorjitter	53.40%	55.96%	54.87%
26	randomcrop224,morpho_erosion,hflip,gray	11.39%	13.02%	12.31%
27	randomcrop224,affine,invert,gaussianblur	11.11%	12.25%	11.25%
28	randomcrop224,morpho_erosion,colorjitter,hflip	10.54%	12.17%	11.17%
29	randomcrop224,affine	48.03%	55.59%	55.44%
30	randomcrop224,morpho_dilation,invert	48.95%	52.67%	53.66%
31	randomcrop224,morpho_erosion,affine,gaussianblur	14.51%	16.56%	14.74%
32	randomcrop224,morpho_erosion,colorjitter,invert	11.48%	13.62%	13.76%
33	randomcrop224,morpho_erosion,morpho_dilation,affine	10.85%	12.12%	11.35%
34	randomcrop224,morpho_dilation,gaussianblur,gray	58.31%	61.34%	61.43%
35	randomcrop224,colorjitter,gaussianblur	59.33%	61.62%	62.31%
36	randomcrop224,morpho_erosion,invert	11.71%	12.68%	12.70%
37	randomcrop224,morpho_erosion,morpho_dilation	58.38%	61.42%	61.49%
38	randomcrop224,affine,hflip	48.76%	55.22%	55.73%
39	randomcrop224,colorjitter,invert	53.93%	56.47%	56.16%
40	randomcrop224,morpho_dilation,colorjitter,hflip	50.11%	52.52%	51.94%
41	randomcrop224	58.84%	60.50%	61.45%
42	randomcrop224,colorjitter,invert,gray	11.23%	12.12%	11.35%
43	randomcrop224,hflip,invert	11.98%	14.19%	13.82%
44	randomcrop224,affine,hflip,gray	11.20%	12.12%	11.35%
45	randomcrop224,colorjitter,invert,gaussianblur	11.00%	12.12%	11.35%
46	randomcrop224,morpho_dilation,gaussianblur	58.85%	62.20%	62.49%
47	randomcrop224,morpho_erosion,invert,gaussianblur	57.16%	59.72%	59.96%
48	randomcrop224,hflip,invert,gaussianblur	11.69%	13.02%	12.11%
49	randomcrop224,morpho_dilation	54.78%	57.29%	56.36%
50	randomcrop224,morpho_erosion,affine,hflip	47.14%	53.08%	53.15%
51	randomcrop224,morpho_erosion,hflip	49.90%	51.61%	51.12%
52	randomcrop224,morpho_dilation,affine,hflip	11.87%	13.37%	11.98%
53	randomcrop224,colorjitter,gray	52.50%	56.90%	55.60%
54	randomcrop224,morpho_dilation,colorjitter,gaussianblur	12.78%	15.46%	14.93%
55	randomcrop224,affine,invert,gray	11.27%	12.78%	12.92%
56	randomcrop224,morpho_erosion,morpho_dilation,	57.94%	61.32%	60.65%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
	gaussianblur			
57	randomcrop224,morpho\_erosion,hflip,invert	53.18%	53.98%	54.38%
58	randomcrop224,morpho\_dilation,gray	11.76%	12.84%	12.13%
59	randomcrop224,affine,colorjitter,hflip	10.88%	12.12%	11.35%
60	randomcrop224,hflip,invert,gray	12.67%	15.27%	14.29%
61	randomcrop224,morpho\_erosion,gray	46.56%	47.90%	48.55%
62	randomcrop224,morpho\_dilation,colorjitter,invert	11.40%	12.96%	12.49%
63	randomcrop224,morpho\_erosion	58.51%	60.66%	60.80%
64	randomcrop224,morpho\_dilation,invert,gaussianblur	11.12%	12.21%	10.96%
65	randomcrop224,affine,gaussianblur,gray	11.83%	13.23%	12.13%
66	randomcrop224,invert	46.54%	47.00%	45.89%
67	randomcrop224,morpho\_erosion,gaussianblur,gray	51.88%	54.94%	55.03%
68	randomcrop224,affine,colorjitter,invert	11.38%	12.49%	11.17%
69	randomcrop224,morpho\_erosion,affine,gray	11.39%	12.12%	11.35%
70	randomcrop224,hflip,gray	47.53%	51.28%	50.76%
71	randomcrop224,morpho\_dilation,hflip,invert	11.22%	12.16%	11.41%
72	randomcrop224,colorjitter,hflip,gaussianblur	11.24%	12.16%	11.27%
73	randomcrop224,morpho\_erosion,affine	55.08%	62.38%	61.47%
74	randomcrop224,morpho\_erosion,colorjitter,gaussianblur	59.51%	62.09%	62.33%
<b>75</b>	<b>randomcrop224,gaussianblur</b>	<b>59.76%</b>	<b>62.77%</b>	<b>61.33%</b>
76	randomcrop224,gray	50.88%	54.51%	54.58%
77	randomcrop224,affine,colorjitter	11.30%	12.12%	11.35%
78	randomcrop224,morpho\_erosion,morpho\_dilation,hflip	11.97%	12.68%	12.02%
79	randomcrop224,affine,hflip,gaussianblur	11.39%	12.14%	11.35%
80	randomcrop224,affine,invert	11.04%	12.12%	11.35%
81	randomcrop224,morpho\_erosion,colorjitter,gray	46.36%	48.50%	49.49%
82	randomcrop224,affine,colorjitter,gaussianblur	12.48%	15.25%	15.19%
83	randomcrop224,hflip,gaussianblur,gray	55.44%	56.51%	57.30%
84	randomcrop224,colorjitter,hflip	50.90%	53.81%	52.90%
85	randomcrop224,morpho\_dilation,colorjitter	51.50%	54.04%	54.23%
86	randomcrop224,affine,gray	11.76%	12.84%	11.92%
87	randomcrop224,hflip,gaussianblur	55.26%	59.05%	58.40%
88	randomcrop224,morpho\_erosion,hflip,gaussianblur	56.04%	57.58%	58.96%
89	randomcrop224,morpho\_erosion,invert,gray	11.08%	12.12%	11.35%
90	randomcrop224,affine,gaussianblur	12.20%	13.29%	12.25%
91	randomcrop224,morpho\_dilation,affine,colorjitter	11.04%	12.12%	11.35%
92	randomcrop224,morpho\_dilation,affine,gaussianblur	51.96%	60.19%	59.08%
93	randomcrop224,invert,gaussianblur	11.44%	12.72%	11.49%

TABLE A.8: Performance of Triplet ResNet-18 Model with triplet loss across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset without backbone

### A.5.3 Triplet Model Using ResNet-18 fine-tuning on ICDAR Dataset With Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_erosion	81.90%	79.53%	78.69%
2	randomcrop224,affine,hflip,gaussianblur	78.97%	78.49%	78.38%
3	randomcrop224,morpho_erosion,morpho_dilation,colorjitter	79.55%	78.98%	78.34%
4	randomcrop224,morpho_erosion,affine,invert	77.95%	78.99%	78.24%
5	randomcrop224,colorjitter,invert,gaussianblur	81.43%	77.88%	77.46%
6	randomcrop224,morpho_dilation,colorjitter	80.56%	79.25%	79.35%
7	randomcrop224,invert,gaussianblur	83.29%	78.37%	77.06%
8	randomcrop224,affine,gaussianblur	81.57%	79.55%	79.20%
9	randomcrop224,morpho_dilation,colorjitter,hflip	79.74%	78.61%	77.73%
10	randomcrop224,morpho_erosion,invert	83.58%	78.65%	78.38%
11	randomcrop224,colorjitter,hflip	80.75%	78.82%	78.22%
12	randomcrop224,morpho_erosion,morpho_dilation	81.84%	79.29%	79.00%
13	randomcrop224,affine,invert,gray	78.85%	79.08%	78.49%
14	randomcrop224,morpho_erosion,affine,colorjitter	79.42%	79.37%	79.08%
15	randomcrop224,morpho_erosion,invert,gray	82.73%	78.12%	77.40%
16	randomcrop224,morpho_dilation,colorjitter,invert	83.05%	76.94%	76.18%
17	randomcrop224,affine,gaussianblur,gray	80.26%	79.19%	79.04%
18	randomcrop224,morpho_dilation,affine,invert	79.58%	78.82%	78.26%
19	randomcrop224,morpho_dilation	85.19%	79.33%	79.18%
20	randomcrop224,affine,hflip,gray	76.44%	77.92%	77.53%
21	randomcrop224,morpho_erosion,colorjitter,gray	79.43%	78.47%	78.02%
22	randomcrop224,colorjitter,gray	79.58%	78.96%	77.96%
23	randomcrop224,morpho_erosion,morpho_dilation,gaussianblur	85.07%	79.37%	78.71%
24	randomcrop224,morpho_dilation,gaussianblur	85.09%	79.33%	78.34%
25	randomcrop224,morpho_dilation,invert,gaussianblur	81.95%	78.74%	78.24%
26	randomcrop224,colorjitter,invert,gray	79.59%	77.00%	76.30%
27	randomcrop224,hflip	85.70%	79.08%	78.85%
28	randomcrop224,morpho_dilation,affine	81.39%	79.74%	79.22%
29	randomcrop224,affine	81.88%	79.94%	79.55%
30	randomcrop224,morpho_dilation,gaussianblur,gray	80.65%	78.99%	78.55%
31	randomcrop224,affine,colorjitter,hflip	78.14%	79.21%	78.22%
32	randomcrop224,invert	85.79%	78.76%	78.10%
33	randomcrop224,gray	81.82%	79.66%	79.28%
34	randomcrop224,morpho_erosion,affine,gray	80.47%	79.33%	79.53%
35	randomcrop224,affine,colorjitter,invert	78.38%	78.25%	78.02%
36	randomcrop224,morpho_erosion,hflip,gaussianblur	84.31%	78.55%	78.24%
37	randomcrop224,morpho_dilation,invert,gray	79.10%	78.16%	76.93%
38	randomcrop224,morpho_dilation,affine,hflip	79.37%	79.02%	77.96%
39	randomcrop224,morpho_erosion,colorjitter,gaussianblur	85.67%	79.78%	79.10%
40	randomcrop224,hflip,invert,gaussianblur	78.64%	77.78%	77.65%
41	randomcrop224,morpho_erosion,morpho_dilation,affine	79.17%	79.64%	78.81%
42	randomcrop224,hflip,gray	83.78%	78.23%	77.83%
43	randomcrop224	80.16%	79.43%	77.73%
44	randomcrop224,affine,gray	80.23%	79.31%	78.38%
45	randomcrop224,gaussianblur,gray	84.01%	79.39%	78.63%
46	randomcrop224,hflip,invert,gray	78.31%	77.73%	77.40%
47	randomcrop224,morpho_erosion,gaussianblur	81.93%	79.04%	78.63%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
48	randomcrop224,morpho\_erosion,morpho\_dilation,hflip	82.48%	78.63%	78.63%
49	randomcrop224,colorjitter	80.74%	79.80%	78.61%
50	randomcrop224,morpho\_erosion,invert,gaussianblur	84.10%	79.04%	77.93%
51	randomcrop224,affine,invert	82.12%	78.82%	78.34%
52	randomcrop224,morpho\_erosion,morpho\_dilation,gray	81.11%	79.39%	78.57%
53	randomcrop224,morpho\_erosion,affine	83.63%	79.51%	78.90%
54	randomcrop224,morpho\_dilation,invert	82.93%	78.86%	78.51%
55	randomcrop224,morpho\_dilation,hflip	80.77%	78.29%	77.57%
56	randomcrop224,hflip,gaussianblur	82.63%	79.10%	78.41%
57	randomcrop224,morpho\_erosion,hflip,gray	81.41%	78.00%	77.65%
58	randomcrop224,colorjitter,hflip,gray	80.81%	77.18%	77.20%
59	randomcrop224,colorjitter,gaussianblur,gray	85.82%	78.98%	78.49%
60	randomcrop224,morpho\_erosion,gaussianblur,gray	81.16%	78.65%	78.04%
61	randomcrop224,hflip,gaussianblur,gray	80.40%	78.70%	78.69%
62	randomcrop224,affine,hflip	79.54%	79.23%	78.41%
63	randomcrop224,colorjitter,gaussianblur	84.25%	79.68%	79.75%
64	randomcrop224,morpho\_dilation,gray	85.75%	78.37%	77.53%
65	randomcrop224,affine,colorjitter,gray	82.79%	79.23%	78.77%
66	randomcrop224,morpho\_dilation,colorjitter,gray	79.64%	78.94%	78.55%
67	randomcrop224,morpho\_erosion,colorjitter,invert	82.88%	77.78%	77.65%
68	randomcrop224,affine,hflip,invert	77.41%	77.71%	77.26%
69	randomcrop224,morpho\_erosion,affine,hflip	80.59%	79.19%	78.88%
70	randomcrop224,morpho\_erosion,hflip	80.82%	78.88%	78.65%
71	randomcrop224,morpho\_erosion,colorjitter	82.48%	79.45%	79.75%
72	randomcrop224,affine,colorjitter	81.39%	79.04%	78.96%
<b>73</b>	<b>randomcrop224,morpho\_dilation,affine,colorjitter</b>	<b>82.85%</b>	<b>80.11%</b>	<b>79.16%</b>
74	randomcrop224,morpho\_dilation,affine,gaussianblur	83.68%	79.96%	79.69%
75	randomcrop224,invert,gaussianblur,gray	80.83%	77.94%	77.18%
76	randomcrop224,affine,invert,gaussianblur	80.52%	78.61%	78.59%
77	randomcrop224,colorjitter,hflip,gaussianblur	78.72%	78.27%	77.32%
78	randomcrop224,morpho\_erosion,gray	84.21%	78.47%	77.57%
79	randomcrop224,morpho\_erosion,morpho\_dilation,invert	83.37%	78.68%	78.41%
80	randomcrop224,morpho\_erosion,affine,gaussianblur	81.02%	79.08%	78.79%
81	randomcrop224,morpho\_erosion,hflip,invert	84.80%	78.18%	77.18%
82	randomcrop224,invert,gray	80.85%	78.49%	77.59%
83	randomcrop224,morpho\_dilation,hflip,gray	79.12%	78.65%	77.85%
84	randomcrop224,hflip,invert	78.06%	77.78%	77.40%
85	randomcrop224,gaussianblur	82.16%	79.35%	79.26%
86	randomcrop224,morpho\_dilation,hflip,invert	81.90%	78.35%	77.55%
87	randomcrop224,morpho\_erosion,colorjitter,hflip	79.92%	77.69%	77.38%
88	randomcrop224,morpho\_dilation,affine,gray	79.59%	79.41%	78.65%
89	randomcrop224,colorjitter,hflip,invert	76.74%	76.92%	76.28%
90	randomcrop224,affine,colorjitter,gaussianblur	79.82%	79.33%	79.06%
91	randomcrop224,morpho\_dilation,hflip,gaussianblur	85.71%	78.94%	77.79%
92	randomcrop224,morpho\_dilation,colorjitter,gaussianblur	81.94%	78.68%	78.43%
93	randomcrop224,colorjitter,invert	79.78%	78.41%	77.65%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
-----	----------------	-----------	----------------	----------

TABLE A.9: Performance of Triplet ResNet-18 Model with triplet loss across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset with backbone

#### A.5.4 Triplet model using ResNet-18 Pre-training on Alpub dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,colorjitter,hflip,invert	99.27%	85.43%	83.68%
2	randomcrop224,hflip,gray	99.15%	85.97%	83.68%
<b>3</b>	<b>randomcrop224,invert,gaussianblur,gray</b>	<b>99.39%</b>	<b>86.94%</b>	<b>85.17%</b>
4	randomcrop224,morpho_dilation,hflip	99.44%	86.26%	85.12%
<b>5</b>	<b>randomcrop224,morpho_dilation,affine,colorjitter</b>	<b>99.33%</b>	<b>87.54%</b>	<b>87.15%</b>
6	randomcrop224, affine, gaussianblur	99.38%	86.60%	86.01%
7	randomcrop224, morpho_dilation, affine	99.18%	86.84%	84.87%
8	randomcrop224, affine, colorjitter	99.28%	86.34%	85.27%

TABLE A.10: Performance of Triplet ResNet-18 Model with triplet loss across various Data Augmentation combinations pertaining on Alpub Dataset

#### A.5.5 Triplet Model Using ResNet-18 fine-tuning on Alpub Dataset Without Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
<b>1</b>	<b>randomcrop224,morpho_dilation,hflip</b>	<b>57.41%</b>	<b>59.66%</b>	<b>59.88%</b>
2	randomcrop224,hflip,gray	54.89%	57.49%	59.06%
3	randomcrop224,colorjitter,hflip,invert	47.08%	42.12%	42.58%
4	randomcrop224,invert,gaussianblur,gray	43.18%	51.03%	52.47%
<b>5</b>	<b>randomcrop224,morpho_dilation,affine,colorjitter</b>	<b>55.21%</b>	<b>63.20%</b>	<b>63.09%</b>
6	randomcrop224, affine, gaussianblur	55.52%	61.62%	61.94%
7	randomcrop224, morpho_dilation, affine	51.45%	58.31%	58.79%
8	randomcrop224, affine, colorjitter	53.48%	59.89%	60.59%

TABLE A.11: Performance of Triplet ResNet-18 Model with triplet loss across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset without backbone

#### A.5.6 Triplet Model Using ResNet-18 fine-tuning on Alpub Dataset With Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
<b>1</b>	<b>randomcrop224,morpho_dilation,hflip</b>	<b>82.47%</b>	<b>78.20%</b>	<b>77.51%</b>

No.	Transform Type	Train Acc	Validation Acc	Test Acc
2	randomcrop224,hflip,gray	79.86%	77.73%	77.03%
3	randomcrop224,colorjitter,hflip,invert	79.63%	77.10%	76.32%
4	randomcrop224,invert,gaussianblur,gray	79.82%	77.63%	76.79%
<b>5</b>	<b>randomcrop224,morpho_dilation,affine,colorjitter</b>	<b>80.97%</b>	<b>79.57%</b>	<b>78.88%</b>
6	randomcrop224, morpho_dilation, affine	80.02%	79.19%	79.47%
7	randomcrop224, affine, gaussianblur	79.46%	79.55%	78.81%
8	randomcrop224, affine, colorjitter	82.90%	79.47%	78.57%

TABLE A.12: Performance of Triplet ResNet-18 Model with triplet loss across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset with backbone

#### A.5.7 Triplet model using ResNet-50 Pre-training on ICDAR dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_erosion,hflip,gray	50.62%	52.59%	53.33%
2	randomcrop224,morpho_erosion,morpho_dilation,gray	51.84%	52.40%	52.04%
3	randomcrop224,morpho_erosion,morpho_dilation,invert	51.63%	52.81%	51.34%
4	randomcrop224,morpho_erosion,colorjitter	82.06%	79.35%	79.90%
5	randomcrop224,gaussianblur	87.31%	80.80%	77.51%
6	randomcrop224,morpho_dilation,affine,gray	51.74%	53.81%	52.04%
7	randomcrop224,morpho_erosion,morpho_dilation,affine	51.17%	52.67%	52.49%
8	randomcrop224,colorjitter,gaussianblur,gray	51.68%	53.49%	53.08%
9	randomcrop224,morpho_dilation,affine,invert	51.85%	52.44%	52.59%
10	randomcrop224,affine,gaussianblur,gray	76.67%	77.37%	75.62%
11	randomcrop224,morpho_erosion,gaussianblur,gray	51.83%	53.28%	52.59%
12	randomcrop224,morpho_erosion,affine,gray	51.73%	52.67%	52.24%
13	randomcrop224,morpho_erosion,colorjitter,gaussianblur	52.05%	54.53%	56.27%
14	randomcrop224,affine,colorjitter,invert	52.36%	52.10%	52.09%
15	randomcrop224,affine,invert,gaussianblur	50.10%	52.24%	49.30%
16	randomcrop224,morpho_erosion,gaussianblur	51.92%	52.59%	54.23%
17	randomcrop224,morpho_dilation,gaussianblur,gray	52.07%	52.20%	52.39%
18	randomcrop224,hflip,invert,gaussianblur	49.88%	53.00%	52.04%
19	randomcrop224,invert,gaussianblur	51.68%	51.77%	49.25%
20	randomcrop224,affine,invert	52.03%	51.77%	52.09%
21	randomcrop224,colorjitter	51.98%	52.71%	53.58%
22	randomcrop224,colorjitter,gray	52.42%	52.87%	52.39%
23	randomcrop224,morpho_erosion,hflip,invert	51.65%	53.16%	51.99%
24	randomcrop224,gaussianblur,gray	52.65%	52.40%	52.19%
25	randomcrop224,morpho_erosion,colorjitter,gray	80.66%	80.11%	80.50%
26	randomcrop224,affine,hflip,gaussianblur	52.11%	52.87%	52.34%
27	randomcrop224,morpho_dilation,colorjitter,gaussianblur	51.32%	52.89%	52.94%
28	randomcrop224,morpho_erosion,invert,gray	50.21%	53.85%	51.44%
29	randomcrop224,affine,colorjitter,hflip	51.91%	52.61%	52.94%
30	randomcrop224,hflip,invert,gray	51.75%	52.85%	52.79%
31	randomcrop224,morpho_erosion,invert,gaussianblur	51.82%	52.53%	53.03%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
32	randomcrop224,morpho\_erosion,colorjitter,hflip	52.00%	53.83%	52.39%
33	randomcrop224,colorjitter,invert,gaussianblur	52.16%	52.83%	53.18%
34	randomcrop224,gray	91.70%	83.48%	81.84%
35	randomcrop224,morpho\_erosion,morpho\_dilation	50.57%	53.04%	52.24%
36	randomcrop224,morpho\_dilation,colorjitter,invert	51.59%	52.30%	51.49%
37	randomcrop224,morpho\_erosion,hflip	52.66%	53.44%	51.19%
38	randomcrop224,morpho\_dilation,affine,colorjitter	52.02%	52.73%	51.74%
39	randomcrop224,invert,gray	52.21%	52.67%	50.90%
40	randomcrop224,morpho\_erosion,morpho\_dilation,colorjitter	51.80%	53.02%	52.14%
41	randomcrop224,morpho\_dilation,colorjitter,gray	51.65%	52.65%	52.89%
42	randomcrop224,morpho\_erosion,colorjitter,invert	51.56%	53.55%	51.94%
43	randomcrop224,hflip,gaussianblur	65.90%	67.63%	65.87%
44	randomcrop224,colorjitter,hflip,invert	50.59%	52.34%	50.50%
45	randomcrop224,colorjitter,hflip,gaussianblur	64.03%	68.19%	65.97%
46	randomcrop224,affine,colorjitter,gray	51.96%	53.26%	52.89%
47	randomcrop224,morpho\_dilation,gray	53.35%	58.37%	60.45%
48	randomcrop224,affine,hflip,invert	51.73%	52.34%	51.29%
49	randomcrop224,hflip,gray	50.28%	51.83%	52.29%
50	randomcrop224,morpho\_dilation,colorjitter	50.36%	52.50%	51.94%
51	randomcrop224,invert	52.34%	52.34%	52.69%
52	randomcrop224,invert,gaussianblur,gray	51.37%	52.73%	49.75%
53	randomcrop224,morpho\_dilation,affine,gaussianblur	82.01%	80.54%	78.31%
54	randomcrop224,morpho\_erosion,invert	51.91%	52.87%	52.04%
55	randomcrop224,morpho\_dilation,affine	82.81%	80.78%	78.36%
56	randomcrop224,hflip,gaussianblur,gray	58.28%	57.66%	56.97%
57	randomcrop224,morpho\_erosion,affine	83.36%	80.45%	78.21%
58	randomcrop224,morpho\_erosion,affine,hflip	51.96%	52.75%	53.33%
59	randomcrop224,affine,gaussianblur	52.08%	51.87%	53.18%
60	randomcrop224,morpho\_erosion,affine,colorjitter	51.86%	52.18%	53.63%
61	randomcrop224,morpho\_erosion,morpho\_dilation,gaussianblur	92.06%	82.64%	82.09%
62	randomcrop224,morpho\_dilation,colorjitter,hflip	51.97%	52.36%	52.04%
63	randomcrop224,affine,colorjitter	52.31%	52.38%	52.79%
64	randomcrop224,affine,hflip	52.11%	52.24%	52.29%
65	randomcrop224,hflip	50.38%	51.81%	50.15%
66	randomcrop224,morpho\_dilation,gaussianblur	93.43%	82.60%	80.05%
67	randomcrop224,morpho\_dilation,hflip,gray	55.31%	58.09%	55.62%
68	randomcrop224,colorjitter,gaussianblur	51.72%	52.97%	53.98%
69	randomcrop224,colorjitter,invert	52.02%	52.22%	53.33%
70	randomcrop224,morpho\_erosion,hflip,gaussianblur	52.58%	52.69%	54.68%
71	randomcrop224,affine	51.83%	51.71%	52.39%
72	randomcrop224,morpho\_dilation,hflip	52.17%	52.24%	52.19%
73	randomcrop224,morpho\_dilation,invert	52.06%	52.52%	50.70%
74	randomcrop224,morpho\_dilation,hflip,invert	51.86%	52.38%	50.65%
<b>75</b>	<b>randomcrop224</b>	<b>94.69%</b>	<b>83.97%</b>	<b>81.99%</b>
76	randomcrop224,colorjitter,hflip	52.71%	52.79%	51.69%
77	randomcrop224,morpho\_dilation,affine,gaussianblur	51.23%	53.34%	52.99%
78	randomcrop224,morpho\_dilation,invert,gaussianblur	51.23%	52.55%	51.54%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
79	randomcrop224,morpho_dilation,invert,gray	51.05%	52.69%	51.39%
80	randomcrop224,colorjitter,hflip,gray	51.66%	53.08%	52.69%
81	randomcrop224,hflip,invert	51.53%	52.08%	52.29%
82	randomcrop224,morpho_erosion	60.45%	64.73%	65.27%
83	randomcrop224,morpho_erosion,gray	51.77%	52.05%	51.79%
84	randomcrop224,morpho_erosion,morpho_dilation,hflip	83.83%	81.52%	80.25%
85	randomcrop224,morpho_erosion,affine,invert	52.10%	52.57%	52.24%
86	randomcrop224,affine,colorjitter,gaussianblur	50.99%	53.16%	54.33%
87	randomcrop224,affine,hflip,gray	50.82%	52.22%	49.05%
88	randomcrop224,colorjitter,invert,gray	51.20%	53.08%	52.19%
89	randomcrop224,morpho_dilation	89.67%	82.89%	81.00%
90	randomcrop224,affine,gray	52.27%	52.59%	52.49%
91	randomcrop224,morpho_dilation,hflip,gaussianblur	61.67%	65.98%	66.32%
92	randomcrop224,morpho_erosion,affine,gaussianblur	74.53%	79.47%	78.26%
93	randomcrop224,affine,invert,gray	50.64%	52.87%	51.29%

TABLE A.13: Performance of Triplet ResNet-50 Model with triplet loss across various Data Augmentation combinations pertaining on ICDAR Dataset

### A.5.8 Triplet Model Using ResNet-50 pretrain on ICDAR dataset fine-tuning on ICDAR Dataset Without Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_dilation,gaussianblur	59.60%	62.15%	61.37%
2	randomcrop224,invert,gray	11.49%	12.66%	11.66%
3	randomcrop224,morpho_erosion,morpho_dilation,colorjitter	11.20%	12.55%	11.47%
4	randomcrop224,morpho_erosion,affine,hflip	15.05%	16.99%	16.11%
5	randomcrop224,affine,colorjitter,gray	10.78%	12.12%	11.35%
6	randomcrop224,colorjitter,invert,gaussianblur	11.19%	12.29%	11.43%
7	randomcrop224,morpho_erosion,colorjitter,gaussianblur	17.97%	21.02%	20.53%
8	randomcrop224,invert	13.17%	14.58%	13.89%
9	randomcrop224,morpho_erosion,morpho_dilation	11.36%	12.49%	10.98%
10	randomcrop224,morpho_erosion,affine,gaussianblur	47.98%	54.32%	53.93%
11	randomcrop224,morpho_erosion,affine,colorjitter	11.79%	12.82%	11.90%
12	randomcrop224,gaussianblur,gray	11.57%	12.72%	12.09%
13	randomcrop224,colorjitter,invert,gray	13.06%	14.82%	13.52%
14	randomcrop224,morpho_dilation,affine,invert	11.36%	12.74%	11.78%
15	randomcrop224,morpho_dilation,colorjitter,hflip	11.45%	12.39%	12.45%
<b>16</b>	<b>randomcrop224</b>	<b>63.33%</b>	<b>64.79%</b>	<b>64.58%</b>
17	randomcrop224,affine,invert	11.29%	12.39%	10.98%
18	randomcrop224,morpho_erosion,hflip,invert	11.34%	12.41%	11.14%
19	randomcrop224,affine,invert,gaussianblur	11.40%	12.59%	12.04%
20	randomcrop224,morpho_erosion,hflip,gray	11.27%	12.16%	11.88%
21	randomcrop224,affine,hflip,gray	11.50%	12.37%	11.59%
22	randomcrop224,morpho_dilation,colorjitter,gray	11.17%	12.80%	12.74%
23	randomcrop224,morpho_erosion,colorjitter,gray	55.31%	58.50%	58.26%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
24	randomcrop224,morpho_dilation,affine,hflip	46.69%	54.04%	53.19%
25	randomcrop224,morpho_dilation,colorjitter,gaussianblur	12.69%	14.33%	13.76%
26	randomcrop224,morpho_erosion,morpho_dilation,hflip	53.24%	56.51%	56.28%
27	randomcrop224,colorjitter,gaussianblur,gray	11.19%	12.16%	11.06%
28	randomcrop224,morpho_dilation,invert	11.40%	12.45%	11.64%
29	randomcrop224,colorjitter,invert	11.60%	12.61%	12.52%
30	randomcrop224,colorjitter,gaussianblur	11.40%	12.21%	11.35%
31	randomcrop224,morpho_dilation,affine	52.45%	59.39%	58.77%
32	randomcrop224,gaussianblur	54.73%	56.59%	56.26%
33	randomcrop224,hflip	12.32%	13.11%	13.60%
34	randomcrop224,colorjitter,hflip	11.01%	12.12%	11.35%
35	randomcrop224,colorjitter,gray	11.50%	12.37%	11.19%
36	randomcrop224,affine,gray	11.19%	12.23%	11.47%
37	randomcrop224,affine,gaussianblur,gray	42.52%	50.30%	49.45%
38	randomcrop224,morpho_erosion,affine	51.30%	57.70%	57.01%
39	randomcrop224,hflip,invert	13.74%	12.98%	12.07%
40	randomcrop224,invert,gaussianblur	15.09%	15.76%	14.56%
41	randomcrop224,morpho_erosion,affine,gray	11.48%	12.31%	11.31%
42	randomcrop224,morpho_erosion,hflip,gaussianblur	11.25%	12.14%	11.68%
43	randomcrop224,affine,invert,gray	10.49%	12.12%	11.35%
44	randomcrop224,morpho_erosion,gaussianblur	11.39%	12.35%	11.51%
45	randomcrop224,affine,gaussianblur	12.00%	13.39%	12.66%
46	randomcrop224,morpho_dilation,colorjitter,invert	11.37%	12.16%	11.33%
47	randomcrop224,hflip,invert,gray	13.03%	11.94%	12.95%
48	randomcrop224,hflip,gaussianblur,gray	21.40%	22.04%	22.41%
49	randomcrop224,morpho_erosion,gray	10.67%	12.12%	11.35%
50	randomcrop224,morpho_dilation,gaussianblur,gray	12.91%	13.41%	12.76%
51	randomcrop224,colorjitter,hflip,gray	13.13%	13.19%	12.41%
52	randomcrop224,morpho_dilation,affine,gray	11.39%	12.43%	11.45%
53	randomcrop224,morpho_dilation,invert,gray	11.40%	12.59%	11.62%
54	randomcrop224,morpho_dilation,gray	18.90%	20.96%	19.26%
55	randomcrop224,colorjitter,hflip,gaussianblur	30.74%	32.43%	31.98%
56	randomcrop224,morpho_erosion,gaussianblur,gray	11.29%	12.55%	11.49%
57	randomcrop224,morpho_erosion,invert	11.11%	12.23%	11.31%
58	randomcrop224,morpho_erosion,morpho_dilation, gaussianblur	60.20%	63.63%	62.97%
59	randomcrop224,morpho_erosion,morpho_dilation,invert	11.33%	12.16%	11.31%
60	randomcrop224,morpho_dilation,hflip,invert	11.69%	13.21%	13.42%
61	randomcrop224,morpho_erosion,colorjitter,invert	11.55%	13.74%	13.05%
62	randomcrop224,morpho_erosion	31.79%	33.71%	32.90%
63	randomcrop224,colorjitter,hflip,invert	12.03%	12.21%	10.67%
64	randomcrop224,affine,colorjitter	12.47%	15.60%	15.46%
65	randomcrop224,morpho_erosion,hflip	16.29%	18.34%	17.14%
66	randomcrop224,affine	11.59%	12.27%	12.72%
67	randomcrop224,affine,hflip	11.41%	12.14%	11.35%
68	randomcrop224,morpho_dilation,colorjitter	13.60%	15.83%	14.48%
69	randomcrop224,morpho_erosion,invert,gray	15.59%	17.05%	15.69%
70	randomcrop224,affine,colorjitter,gaussianblur	11.32%	12.19%	11.29%
71	randomcrop224,invert,gaussianblur,gray	13.51%	14.62%	13.33%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
72	randomcrop224,morpho_dilation,hflip,gray	18.73%	20.40%	18.94%
73	randomcrop224,morpho_dilation,affine,colorjitter	12.56%	15.27%	14.03%
74	randomcrop224,morpho_dilation,invert,gaussianblur	12.29%	14.07%	13.99%
75	randomcrop224,morpho_erosion,affine,invert	11.30%	12.33%	11.57%
76	randomcrop224,morpho_erosion,colorjitter,hflip	11.39%	12.62%	11.43%
77	randomcrop224,affine,colorjitter,invert	11.35%	12.17%	11.31%
78	randomcrop224,affine,hflip,invert	13.27%	16.11%	15.11%
79	randomcrop224,morpho_dilation,hflip,gaussianblur	32.58%	33.49%	32.99%
80	randomcrop224,morpho_erosion,colorjitter	50.95%	54.12%	53.66%
81	randomcrop224,morpho_erosion,morpho_dilation,affine	11.24%	12.31%	11.37%
82	randomcrop224,morpho_dilation	55.53%	58.62%	58.40%
83	randomcrop224,colorjitter	11.37%	12.14%	11.33%
84	randomcrop224,morpho_dilation,hflip	13.59%	14.60%	14.29%
85	randomcrop224,hflip,gray	11.24%	12.55%	12.21%
86	randomcrop224,morpho_dilation,affine,gaussianblur	12.26%	13.74%	13.19%
87	randomcrop224,hflip,invert,gaussianblur	11.58%	12.37%	12.33%
88	randomcrop224,morpho_erosion,invert,gaussianblur	11.18%	12.14%	11.33%
89	randomcrop224,morpho_erosion,morpho_dilation,gray	11.40%	12.12%	11.23%
90	randomcrop224,affine,colorjitter,hflip	12.17%	13.37%	12.97%
91	randomcrop224,hflip,gaussianblur	36.22%	38.13%	37.57%
92	randomcrop224,gray	60.24%	62.56%	63.15%
93	randomcrop224,affine,hflip,gaussianblur	12.10%	12.62%	13.23%

TABLE A.14: Performance of Triplet ResNet-50 Model with triplet loss across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset without backbone

### A.5.9 Triplet Model Using ResNet-50 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset With Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,colorjitter,hflip,invert	76.59%	76.36%	75.46%
2	randomcrop224,gaussianblur,gray	80.43%	78.68%	77.65%
3	randomcrop224,morpho_dilation,affine,colorjitter	77.38%	78.72%	77.63%
4	randomcrop224,hflip,invert	78.39%	77.75%	76.59%
5	randomcrop224,morpho_erosion,affine,hflip	73.22%	76.61%	76.38%
6	randomcrop224,morpho_erosion,affine,invert	76.73%	77.65%	77.34%
7	randomcrop224,morpho_dilation,invert	80.27%	78.51%	77.89%
8	randomcrop224,colorjitter,gray	80.93%	78.45%	76.89%
9	randomcrop224,morpho_dilation,affine	80.78%	78.82%	77.63%
10	randomcrop224,morpho_dilation,colorjitter,gray	77.80%	78.57%	77.51%
11	randomcrop224,morpho_erosion,hflip	78.37%	77.39%	77.53%
12	randomcrop224,morpho_erosion,morpho_dilation,affine	77.91%	78.39%	77.20%
13	randomcrop224,morpho_erosion,hflip,invert	76.81%	76.83%	75.99%
14	randomcrop224,affine,colorjitter,gaussianblur	74.69%	77.00%	76.67%
15	randomcrop224,morpho_dilation,hflip,gaussianblur	79.04%	77.61%	76.95%
16	randomcrop224,morpho_erosion,affine,gaussianblur	79.38%	79.10%	78.43%
17	randomcrop224,affine,invert,gaussianblur	78.56%	79.02%	77.75%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
18	randomcrop224,colorjitter,hflip	77.40%	76.83%	76.48%
19	randomcrop224,affine,hflip,gray	72.23%	75.26%	74.83%
20	randomcrop224,morpho_erosion,invert,gaussianblur	80.83%	78.29%	77.59%
21	randomcrop224,morpho_dilation,hflip,invert	78.03%	77.28%	76.75%
22	randomcrop224,hflip,gaussianblur	79.58%	77.51%	77.20%
23	randomcrop224,affine,hflip	74.96%	76.65%	76.54%
24	randomcrop224,morpho_dilation,colorjitter,hflip	76.39%	76.26%	76.50%
25	randomcrop224,morpho_dilation	83.79%	78.68%	78.00%
26	randomcrop224,invert	81.52%	78.27%	77.91%
27	randomcrop224,morpho_erosion,colorjitter,gray	79.48%	78.37%	77.96%
28	randomcrop224,affine,hflip,gaussianblur	75.83%	76.98%	76.75%
29	randomcrop224,hflip	79.10%	77.37%	76.95%
30	randomcrop224,morpho_erosion,invert,gray	77.24%	77.28%	76.83%
31	randomcrop224,morpho_erosion,colorjitter	78.84%	78.82%	77.55%
32	randomcrop224	83.29%	78.82%	78.81%
33	randomcrop224,affine,colorjitter	76.89%	78.41%	78.45%
34	randomcrop224,colorjitter,gaussianblur	80.90%	78.90%	78.69%
35	randomcrop224,affine,gaussianblur,gray	76.27%	77.92%	77.42%
36	randomcrop224,affine,hflip,invert	72.46%	76.43%	75.32%
37	randomcrop224,colorjitter,invert,gray	78.13%	77.53%	77.14%
38	randomcrop224,morpho_dilation,invert,gray	76.91%	77.78%	77.18%
39	randomcrop224,hflip,gaussianblur,gray	75.52%	75.79%	75.46%
40	randomcrop224,colorjitter	82.18%	78.67%	77.87%
41	randomcrop224,gray	80.53%	78.20%	77.93%
42	randomcrop224,morpho_dilation,colorjitter,invert	79.39%	77.43%	76.91%
43	randomcrop224,invert,gaussianblur	78.46%	78.25%	77.77%
44	randomcrop224,affine	77.70%	78.20%	77.63%
45	randomcrop224,affine,invert,gray	74.39%	76.53%	76.75%
46	randomcrop224,morpho_erosion,gaussianblur	83.38%	78.70%	77.65%
47	randomcrop224,gaussianblur	81.91%	78.16%	77.83%
48	randomcrop224,morpho_dilation,affine,gray	77.02%	78.12%	77.77%
49	randomcrop224,affine,colorjitter,hflip	74.43%	76.94%	76.16%
50	randomcrop224,affine,gray	77.09%	78.00%	77.22%
51	randomcrop224,affine,colorjitter,gray	75.50%	77.55%	77.26%
52	randomcrop224,morpho_dilation,gray	78.29%	77.96%	77.22%
53	randomcrop224,invert,gaussianblur,gray	77.78%	78.08%	77.08%
54	randomcrop224,morpho_dilation,hflip,gray	76.17%	76.88%	76.32%
55	randomcrop224,affine,gaussianblur	78.70%	78.94%	77.93%
56	randomcrop224,morpho_erosion,affine,gray	76.02%	79.02%	77.87%
57	randomcrop224,morpho_erosion,colorjitter,hflip	77.89%	77.49%	76.59%
58	randomcrop224,morpho_erosion,gray	79.94%	78.76%	77.77%
59	randomcrop224,morpho_dilation,affine,invert	76.22%	77.92%	76.61%
60	randomcrop224,invert,gray	78.11%	77.98%	77.77%
61	randomcrop224,morpho_dilation,gaussianblur,gray	81.76%	78.53%	78.06%
<b>62</b>	<b>randomcrop224,morpho_erosion,morpho_dilation,gaussianblur</b>	<b>83.70%</b>	<b>79.29%</b>	<b>78.22%</b>
63	randomcrop224,morpho_erosion,morpho_dilation,hflip	78.83%	77.33%	77.06%
64	randomcrop224,colorjitter,hflip,gray	77.74%	77.18%	76.20%
65	randomcrop224,colorjitter,invert	79.16%	77.22%	77.16%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
66	randomcrop224,morpho\_erosion,morpho\_dilation,invert	78.97%	77.98%	77.69%
67	randomcrop224,morpho\_erosion,morpho\_dilation,gray	80.08%	79.15%	78.12%
68	randomcrop224,morpho\_dilation,affine,hflip	74.24%	76.47%	76.16%
69	randomcrop224,colorjitter,hflip,gaussianblur	75.86%	76.20%	74.83%
70	randomcrop224,morpho\_erosion,hflip,gaussianblur	77.80%	77.76%	76.59%
71	randomcrop224,hflip,gray	73.61%	75.85%	74.81%
72	randomcrop224,morpho\_erosion,morpho\_dilation	80.70%	79.12%	78.36%
73	randomcrop224,morpho\_erosion,morpho\_dilation,colorjitter	80.95%	79.04%	78.20%
74	randomcrop224,colorjitter,invert,gaussianblur	77.91%	78.04%	77.42%
75	randomcrop224,morpho\_erosion,affine	80.09%	78.67%	78.55%
76	randomcrop224,morpho\_erosion,gaussianblur,gray	79.47%	78.23%	78.14%
77	randomcrop224,morpho\_erosion,invert	81.83%	79.00%	78.40%
78	randomcrop224,morpho\_erosion,hflip,gray	75.87%	76.45%	75.91%
79	randomcrop224,affine,colorjitter,invert	76.50%	78.04%	77.40%
80	randomcrop224,colorjitter,gaussianblur,gray	79.39%	78.57%	77.71%
81	randomcrop224,hflip,invert,gaussianblur	76.16%	75.81%	74.93%
82	randomcrop224,affine,invert	77.21%	77.88%	77.12%
83	randomcrop224,morpho\_erosion,affine,colorjitter	78.76%	79.23%	78.61%
84	randomcrop224,morpho\_dilation,affine,gaussianblur	78.22%	78.35%	78.10%
85	randomcrop224,morpho\_dilation,colorjitter,gaussianblur	82.25%	78.47%	78.36%
86	randomcrop224,morpho\_erosion,colorjitter,invert	79.36%	77.28%	76.71%
87	randomcrop224,morpho\_erosion	81.29%	79.06%	77.95%
88	randomcrop224,morpho\_dilation,colorjitter	80.90%	79.15%	78.14%
89	randomcrop224,morpho\_dilation,gaussianblur	84.04%	78.68%	77.55%
90	randomcrop224,morpho\_erosion,colorjitter,gaussianblur	81.10%	78.12%	77.87%
91	randomcrop224,morpho\_dilation,hflip	78.99%	77.73%	76.30%
92	randomcrop224,morpho\_dilation,invert,gaussianblur	81.39%	78.70%	77.65%
93	randomcrop224,hflip,invert,gray	73.62%	75.67%	74.60%

TABLE A.15: Performance of Triplet ResNet-50 Model with triplet loss across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset with backbone

#### A.5.10 Triplet model using ResNet-50 Pre-training on Alpub dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho\_dilation,hflip	97.75%	88.55%	87.31%
2	randomcrop224,hflip,gray	97.74%	89.25%	87.46%
3	randomcrop224,colorjitter,hflip,invert	96.57%	87.04%	86.51%
4	<b>randomcrop224,invert,gaussianblur,gray</b>	<b>98.17%</b>	<b>89.74%</b>	<b>89.26%</b>
5	randomcrop224,morpho\_dilation,affine,colorjitter	97.49%	89.65%	89.21%
6	randomcrop224, morpho\_dilation, affine	97.05%	89.99%	89.36%
7	<b>randomcrop224, affine, gaussianblur</b>	<b>97.52%</b>	<b>90.45%</b>	<b>89.76%</b>
8	randomcrop224, affine, colorjitter	96.96%	90.07%	88.86%

No	Transform Type	Train Acc	Validation Acc	Test Acc
----	----------------	-----------	----------------	----------

TABLE A.16: Performance of Triplet ResNet-50 Model with triplet loss across various Data Augmentation combinations pertaining on Alpub Dataset

#### A.5.11 Triplet Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset Without Backbone

No	Transform Type	Train Acc	Validation Acc	Test Acc
1	randomcrop224,colorjitter,hflip,invert	55.76%	57.37%	58.32%
<b>2</b>	<b>randomcrop224,hflip,gray</b>	<b>61.83%</b>	<b>63.38%</b>	<b>63.80%</b>
3	randomcrop224,morpho_dilation,hflip	60.65%	61.48%	62.29%
4	randomcrop224,invert,gaussianblur,gray	54.64%	63.01%	62.78%
5	randomcrop224,morpho_dilation,affine,colorjitter	60.05%	65.18%	66.13%
6	randomcrop224,morpho_dilation,affine	60.07%	65.59%	65.36%
7	randomcrop224, affine, gaussianblur	62.78%	67.12%	67.03%
<b>8</b>	<b>randomcrop224,affine,colorjitter</b>	<b>62.58%</b>	<b>67.35%</b>	<b>67.12%</b>

TABLE A.17: Performance of Triplet ResNet-50 Model with triplet loss across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset without backbone

#### A.5.12 Triplet Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset With Backbone

No	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,hflip,gray	77.56%	77.12%	76.20%
2	randomcrop224,colorjitter,hflip,invert	75.88%	75.89%	74.60%
<b>3</b>	<b>randomcrop224,invert,gaussianblur,gray</b>	<b>77.33%</b>	<b>77.90%</b>	<b>77.03%</b>
4	randomcrop224,morpho_dilation,hflip	79.29%	77.43%	76.79%
<b>5</b>	<b>randomcrop224,morpho_dilation,affine,colorjitter</b>	<b>78.51%</b>	<b>79.17%</b>	<b>78.24%</b>
6	randomcrop224,morpho_dilation,affine	79.31%	78.67%	78.36%
7	randomcrop224,affine,gaussianblur	79.44%	78.86%	78.71%
8	randomcrop224,affine,colorjitter	78.68%	78.70%	78.71%

TABLE A.18: Performance of Triplet ResNet-50 Model with triplet loss across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset with backbone

### A.6 SimCLR model Using ResNet-18 and ResNet-50 Networks

#### A.6.1 SimCLR model using ResNet-18 Pre-training on ICDAR dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop198,morpho\_erosion,morpho\_dilation,invert	99.97%	99.99%	99.96%
2	randomcrop198,morpho\_erosion,gray	99.99%	99.98%	99.95%
3	randomcrop198,morpho\_dilation,affine,gaussianblur	99.84%	99.91%	99.83%
4	randomcrop198,colorjitter,invert,gaussianblur	99.97%	99.98%	99.88%
5	randomcrop198,hflip,invert,gaussianblur	99.93%	99.99%	99.96%
6	randomcrop198,affine,colorjitter,hflip	99.85%	99.79%	99.79%
7	randomcrop198,affine,colorjitter	99.88%	99.90%	99.87%
8	randomcrop198,morpho\_erosion,morpho\_dilation,hflip	99.83%	99.95%	99.83%
9	randomcrop198,morpho\_erosion,invert,gray	99.95%	99.95%	99.93%
10	randomcrop198,invert	99.96%	99.99%	99.95%
11	randomcrop198,hflip,invert,gray	99.86%	99.95%	99.90%
12	randomcrop198,morpho\_dilation,hflip	99.93%	99.98%	99.92%
13	randomcrop198,morpho\_erosion,morpho\_dilation	99.96%	99.98%	99.95%
14	randomcrop198,affine	99.83%	99.88%	99.79%
15	randomcrop198,morpho\_erosion,morpho\_dilation,affine	99.87%	99.90%	99.79%
16	randomcrop198,affine,gaussianblur,gray	99.80%	99.87%	99.69%
17	randomcrop198,morpho\_erosion,morpho\_dilation, gaussianblur	99.96%	99.99%	99.94%
18	randomcrop198,morpho\_dilation,gaussianblur	99.96%	99.99%	99.96%
19	randomcrop198,affine,colorjitter,gaussianblur	99.86%	99.89%	99.73%
20	randomcrop198,morpho\_dilation,invert,gaussianblur	99.99%	99.97%	99.98%
21	randomcrop198,morpho\_erosion,colorjitter,gaussianblur	99.96%	99.98%	99.93%
22	randomcrop198,morpho\_dilation,affine,invert	99.88%	99.95%	99.92%
23	randomcrop198,colorjitter,hflip	99.91%	99.96%	99.89%
24	randomcrop198,morpho\_dilation,colorjitter,hflip	99.92%	99.96%	99.92%
25	randomcrop198,hflip,gaussianblur	99.96%	99.98%	99.92%
26	randomcrop198,morpho\_erosion,hflip,invert	99.94%	99.95%	99.89%
27	randomcrop198,morpho\_dilation,colorjitter,gray	99.97%	99.97%	99.94%
28	randomcrop198,morpho\_erosion,hflip	99.92%	99.98%	99.90%
29	randomcrop198,affine,invert	99.88%	99.88%	99.87%
30	randomcrop198,invert,gaussianblur,gray	99.95%	99.97%	99.95%
31	<b>randomcrop198,gaussianblur</b>	<b>99.94%</b>	<b>100.00%</b>	<b>99.91%</b>
32	randomcrop198,morpho\_dilation,colorjitter,gaussianblur	99.97%	99.96%	99.91%
33	randomcrop198,affine,colorjitter,gray	99.86%	99.85%	99.82%
34	randomcrop198,morpho\_erosion,affine	99.82%	99.89%	99.77%
35	randomcrop198,affine,hflip	99.79%	99.88%	99.78%
36	randomcrop198,hflip,gray	99.94%	99.97%	99.85%
37	randomcrop198,morpho\_dilation,colorjitter,invert	99.96%	99.97%	99.91%
38	randomcrop198,morpho\_dilation,affine,gray	99.90%	99.89%	99.87%
39	randomcrop198,morpho\_dilation,affine,colorjitter	99.89%	99.88%	99.76%
40	randomcrop198,gray	99.96%	99.99%	99.91%
41	randomcrop198,morpho\_erosion,gaussianblur	99.98%	99.99%	99.97%
42	randomcrop198,colorjitter,gray	99.95%	99.97%	99.89%
43	randomcrop198,colorjitter,hflip,invert	99.92%	99.95%	99.87%
44	randomcrop198,morpho\_erosion,invert	99.99%	99.99%	99.94%
45	randomcrop198,affine,hflip,invert	99.80%	99.91%	99.74%
46	randomcrop198,morpho\_erosion,gaussianblur,gray	99.98%	99.99%	99.91%
47	randomcrop198,morpho\_dilation	99.96%	99.98%	99.99%
48	randomcrop198,morpho\_erosion,colorjitter,gray	99.95%	99.94%	99.87%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
49	randomcrop198,morpho\_erosion,affine,gray	99.79%	99.84%	99.70%
50	randomcrop198,invert,gaussianblur	99.95%	99.98%	99.78%
51	randomcrop198,morpho\_erosion,affine,colorjitter	99.85%	99.87%	99.86%
52	randomcrop198,morpho\_dilation,hflip,invert	99.93%	99.94%	99.89%
53	randomcrop198	99.89%	99.97%	99.91%
54	randomcrop198,hflip,invert	99.92%	99.95%	99.93%
55	randomcrop198,morpho\_erosion,hflip,gaussianblur	99.92%	99.96%	99.98%
56	randomcrop198,affine,hflip,gaussianblur	99.84%	99.84%	99.77%
57	randomcrop198,invert,gray	99.97%	99.99%	99.91%
58	randomcrop198,morpho\_dilation,hflip,gray	99.93%	99.97%	99.89%
59	randomcrop198,morpho\_dilation,invert,gray	99.94%	99.95%	99.79%
60	randomcrop198,morpho\_dilation,gray	99.98%	99.99%	99.97%
61	randomcrop198,morpho\_erosion,hflip,gray	99.91%	99.97%	99.84%
62	randomcrop198,affine,hflip,gray	99.84%	99.89%	99.85%
63	randomcrop198,colorjitter	99.97%	99.99%	99.88%
64	randomcrop198,morpho\_erosion	99.96%	100.00%	99.94%
65	randomcrop198,morpho\_dilation,hflip,gaussianblur	99.95%	99.96%	99.88%
66	randomcrop198,colorjitter,gaussianblur,gray	99.90%	99.98%	99.80%
67	randomcrop198,gaussianblur,gray	99.94%	99.98%	99.91%
68	randomcrop198,colorjitter,invert,gray	99.97%	99.94%	99.89%
69	randomcrop198,affine,gray	99.84%	99.91%	99.80%
70	randomcrop198,morpho\_dilation,gaussianblur,gray	99.97%	99.96%	99.92%
71	randomcrop198,colorjitter,hflip,gaussianblur	99.94%	99.97%	99.94%
72	randomcrop198,colorjitter,invert	99.96%	99.96%	99.96%
73	randomcrop198,morpho\_erosion,affine,invert	99.91%	99.90%	99.90%
74	randomcrop198,hflip	99.92%	99.97%	99.90%
75	randomcrop198,morpho\_dilation,affine,hflip	99.84%	99.88%	99.85%
76	randomcrop198,affine,colorjitter,invert	99.82%	99.82%	99.77%
77	randomcrop198,colorjitter,hflip,gray	99.97%	99.92%	99.91%
78	randomcrop198,morpho\_erosion,morpho\_dilation,gray	99.95%	99.98%	99.88%
79	randomcrop198,morpho\_erosion,colorjitter	99.96%	99.97%	99.93%
80	randomcrop198,colorjitter,gaussianblur	99.99%	99.99%	99.96%
81	randomcrop198,morpho\_erosion,colorjitter,invert	99.97%	99.97%	99.94%
82	randomcrop198,hflip,gaussianblur,gray	99.94%	99.98%	99.83%
83	randomcrop198,morpho\_erosion,affine,hflip	99.80%	99.88%	99.76%
84	randomcrop198,morpho\_dilation,affine	99.89%	99.88%	99.91%
85	randomcrop198,morpho\_erosion,affine,gaussianblur	99.80%	99.90%	99.83%
86	randomcrop198,morpho\_erosion,colorjitter,hflip	99.90%	99.94%	99.88%
87	randomcrop198,affine,invert,gaussianblur	99.88%	99.93%	99.86%
88	randomcrop198,affine,invert,gray	99.88%	99.90%	99.86%
89	randomcrop198,morpho\_dilation,colorjitter	99.96%	100.00%	99.96%
90	randomcrop198,morpho\_dilation,invert	99.96%	99.99%	99.98%
91	randomcrop198,affine,gaussianblur	99.86%	99.88%	99.63%
92	randomcrop198,morpho\_erosion,morpho\_dilation, colorjitter	99.96%	99.98%	99.87%
93	randomcrop198,morpho\_erosion,invert,gaussianblur	99.98%	99.99%	99.94%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
-----	----------------	-----------	----------------	----------

TABLE A.19: Performance of SimCLR model ResNet-18 architecture across various Data Augmentation combinations pertaining on ICDAR Dataset

### A.6.2 SimCLR Model Using ResNet-18 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset Without Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,colorjitter,invert	30.24%	30.38%	29.65%
2	randomcrop224,morpho_dilation,affine,hflip	22.8%	22.25%	22.45%
3	randomcrop224,colorjitter,gray	30.04%	30.93%	28.53%
4	randomcrop224,morpho_erosion,invert,gaussianblur	30.52%	33.22%	30.65%
5	randomcrop224,morpho_dilation,colorjitter,invert	34.26%	34.55%	32.8%
6	randomcrop224,affine,invert,gray	29.63%	31.26%	29.9%
7	randomcrop224,morpho_erosion	24.73%	25.09%	24.64%
8	randomcrop224,affine,colorjitter,hflip	30.85%	32.3%	31.1%
9	randomcrop224,colorjitter	30.13%	30.57%	28.24%
10	randomcrop224,morpho_dilation,colorjitter,gaussianblur	31.96%	32.65%	30.43%
11	randomcrop224,morpho_erosion,colorjitter,hflip	31.26%	31.88%	30.33%
12	randomcrop224,morpho_dilation,affine	22.72%	22.98%	21.88%
13	randomcrop224,hflip,invert	28.08%	28.48%	27.71%
14	randomcrop224,invert,gaussianblur	31.83%	32.67%	30.59%
15	randomcrop224,affine	21.26%	22.12%	20.47%
16	randomcrop224	23.63%	23.64%	23.25%
17	randomcrop224,hflip,gaussianblur	23.94%	24.13%	22.84%
18	randomcrop224,morpho_erosion,hflip,invert	29.78%	30.34%	29.49%
19	randomcrop224,hflip,invert,gaussianblur	28.01%	28.36%	27.81%
20	randomcrop224,morpho_dilation,invert,gray	29.66%	30.42%	29.26%
21	randomcrop224,morpho_dilation,affine,gaussianblur	22.95%	24.27%	23.09%
22	randomcrop224,affine,gray	26.39%	26.99%	26.42%
23	randomcrop224,morpho_erosion,colorjitter,gray	33.07%	33.78%	32.43%
24	randomcrop224,affine,gaussianblur,gray	26.13%	27.66%	25.6%
25	randomcrop224,colorjitter,gaussianblur	30.81%	30.75%	29.45%
26	randomcrop224,morpho_dilation,hflip,invert	29.46%	29.4%	28.71%
27	<b>randomcrop224,morpho_erosion,colorjitter,invert</b>	<b>35.58%</b>	<b>37.42%</b>	<b>35.73%</b>
28	randomcrop224,morpho_dilation,colorjitter,gray	31.03%	31.28%	29.51%
29	randomcrop224,colorjitter,hflip,gray	30.64%	31.61%	29.65%
30	randomcrop224,morpho_erosion,affine,gaussianblur	23.97%	23.82%	22.88%
31	randomcrop224,morpho_dilation,gaussianblur,gray	27.29%	28.46%	26.83%
32	randomcrop224,colorjitter,hflip,invert	32.52%	33.76%	32.52%
33	randomcrop224,morpho_dilation,colorjitter,hflip	32.09%	32.94%	31.78%
34	randomcrop224,morpho_erosion,hflip,gaussianblur	26.53%	27.64%	25.46%
35	randomcrop224,affine,invert	31.78%	32.26%	32.05%
36	randomcrop224,affine,colorjitter,gaussianblur	31.99%	34.29%	32.04%
37	randomcrop224,morpho_erosion,hflip,gray	27.33%	26.91%	26.89%
38	randomcrop224,morpho_erosion,gray	27.59%	28.34%	26.67%
39	randomcrop224,affine,hflip,gaussianblur	20.24%	20.85%	21.02%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
40	randomcrop224,morpho\_erosion,hflip	24.74%	24.58%	23.7%
41	randomcrop224,morpho\_dilation	26.63%	27.46%	26.13%
42	randomcrop224,morpho\_erosion,invert	30.06%	30.57%	30.37%
43	randomcrop224,morpho\_erosion,colorjitter	29.94%	30.22%	28.92%
44	randomcrop224,morpho\_erosion,morpho\_dilation	23.72%	24.02%	23.46%
45	randomcrop224,hflip	24.05%	24.09%	23.15%
46	randomcrop224,morpho\_erosion,gaussianblur	26.24%	27.46%	26.34%
47	randomcrop224,morpho\_dilation,invert	30.17%	30.2%	29.32%
48	randomcrop224,colorjitter,gaussianblur,gray	30.43%	31.08%	29.8%
49	randomcrop224,morpho\_dilation,gaussianblur	25.94%	26.25%	25.91%
50	randomcrop224,morpho\_dilation,affine,invert	29.43%	30.93%	30.27%
51	randomcrop224,hflip,gaussianblur,gray	27.19%	27.21%	25.93%
52	randomcrop224,invert	27.63%	28.69%	26.77%
53	randomcrop224,morpho\_erosion,morpho\_dilation,colorjitter	32.53%	33.59%	31.66%
54	randomcrop224,morpho\_erosion,colorjitter,gaussianblur	35.88%	37.37%	35.52%
55	randomcrop224,morpho\_dilation,hflip,gaussianblur	24.27%	24.62%	23.97%
56	randomcrop224,colorjitter,invert,gaussianblur	32.35%	33.04%	31.51%
57	randomcrop224,morpho\_dilation,gray	26.85%	27.7%	27.03%
58	randomcrop224,gaussianblur,gray	27.91%	28.15%	27.3%
59	randomcrop224,morpho\_erosion,morpho\_dilation,invert	30.45%	32.3%	30.76%
60	randomcrop224,morpho\_erosion,affine,colorjitter	33.4%	35.29%	33.6%
61	randomcrop224,affine,hflip,invert	26.83%	28.13%	26.99%
62	randomcrop224,colorjitter,hflip,gaussianblur	31.11%	32.18%	31.02%
63	randomcrop224,gray	29.27%	28.87%	28.36%
64	randomcrop224,affine,colorjitter	31.03%	32.53%	32.27%
65	randomcrop224,invert,gray	28.67%	29.16%	27.93%
66	randomcrop224,morpho\_dilation,invert,gaussianblur	32.47%	34.17%	32.35%
67	randomcrop224,morpho\_erosion,gaussianblur,gray	28.83%	29.28%	28.51%
68	randomcrop224,morpho\_dilation,hflip	26.18%	26.38%	25.64%
69	randomcrop224,morpho\_erosion,morpho\_dilation,gaussianblur	26.01%	26.76%	24.91%
70	randomcrop224,affine,invert,gaussianblur	27.97%	28.66%	28.22%
71	randomcrop224,morpho\_erosion,affine,hflip	22.58%	23.57%	22.56%
72	randomcrop224,colorjitter,invert,gray	31.06%	32.06%	30.18%
73	randomcrop224,hflip,gray	25.71%	26.07%	24.85%
74	randomcrop224,morpho\_dilation,affine,colorjitter	33.92%	35.74%	35.38%
75	randomcrop224,morpho\_erosion,morpho\_dilation,gray	27.66%	28.34%	27.06%
76	randomcrop224,morpho\_dilation,hflip,gray	28.78%	29.26%	28.34%
77	randomcrop224,affine,gaussianblur	21.0%	22.16%	20.23%
78	randomcrop224,morpho\_erosion,affine,gray	27.81%	28.62%	27.57%
79	randomcrop224,morpho\_dilation,colorjitter	29.96%	30.81%	29.37%
80	randomcrop224,affine,colorjitter,gray	32.51%	34.33%	33.42%
81	randomcrop224,colorjitter,hflip	30.13%	30.4%	29.28%
82	randomcrop224,morpho\_erosion,affine,invert	29.07%	29.83%	29.96%
83	randomcrop224,affine,hflip,gray	25.91%	27.15%	25.85%
84	randomcrop224,morpho\_erosion,invert,gray	28.41%	28.77%	27.08%
85	randomcrop224,morpho\_erosion,affine	23.1%	22.71%	22.43%
86	randomcrop224,morpho\_dilation,affine,gray	27.87%	28.73%	27.83%
87	randomcrop224,morpho\_erosion,morpho\_dilation,affine	23.71%	23.64%	23.5%
88	randomcrop224,affine,hflip	22.1%	22.39%	20.59%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
89	randomcrop224,hflip,invert,gray	27.91%	28.6%	27.65%
90	randomcrop224,gaussianblur	25.04%	25.15%	23.82%
91	randomcrop224,invert,gaussianblur,gray	29.04%	28.83%	28.08%
92	randomcrop224,morpho\_erosion,morpho\_dilation,hflip	24.75%	25.21%	24.4%
93	randomcrop224,affine,colorjitter,invert	34.25%	35.39%	35.69%

TABLE A.20: Performance of SimCLR model ResNet-18 architecture across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset without backbone

### A.6.3 SimCLR Model Using ResNet-18 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset With Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho\_erosion,morpho\_dilation,colorjitter	84.31%	80.05%	79.84%
2	randomcrop224,colorjitter	85.09%	80.05%	79.82%
3	randomcrop224,hflip,invert	80.39%	78.63%	78.86%
4	randomcrop224,affine,colorjitter,gray	82.11%	79.82%	80.16%
5	randomcrop224,hflip	81.52%	79.19%	78.43%
6	randomcrop224,morpho\_erosion,hflip	81.44%	78.92%	79.24%
7	randomcrop224,morpho\_erosion,colorjitter	84.32%	80.0%	80.12%
8	randomcrop224,morpho\_dilation,hflip,gray	79.61%	78.59%	77.93%
9	randomcrop224,invert,gaussianblur,gray	81.34%	79.12%	78.98%
10	<b>randomcrop224,morpho\_erosion,affine</b>	<b>83.24%</b>	<b>81.03%</b>	<b>80.23%</b>
11	randomcrop224,colorjitter,gray	81.06%	79.19%	78.86%
12	randomcrop224,morpho\_erosion,morpho\_dilation,invert	83.84%	80.05%	80.14%
13	randomcrop224,affine,colorjitter	82.53%	80.88%	80.0%
14	randomcrop224,morpho\_dilation,colorjitter,gaussianblur	82.23%	80.13%	79.96%
15	randomcrop224,hflip,gaussianblur	80.96%	78.94%	78.02%
16	randomcrop224,morpho\_erosion,morpho\_dilation,affine	81.77%	80.5%	80.2%
17	randomcrop224,morpho\_dilation,colorjitter,gray	84.28%	79.92%	79.43%
18	randomcrop224,gaussianblur	83.46%	80.27%	80.41%
19	randomcrop224,affine,colorjitter,invert	81.4%	80.47%	80.14%
20	randomcrop224,morpho\_dilation,affine	83.09%	80.66%	80.8%
21	randomcrop224,affine,gaussianblur	84.71%	80.45%	80.45%
22	randomcrop224,invert	82.2%	80.02%	78.96%
23	randomcrop224,affine,invert	85.54%	80.45%	80.1%
24	randomcrop224,morpho\_erosion,morpho\_dilation,hflip	81.18%	79.57%	78.47%
25	randomcrop224,affine,colorjitter,gaussianblur	82.16%	80.96%	80.18%
26	randomcrop224,hflip,gray	80.97%	79.06%	78.73%
27	randomcrop224,affine,gaussianblur,gray	81.43%	80.27%	79.63%
28	randomcrop224,morpho\_dilation,gaussianblur,gray	84.85%	80.19%	79.02%
29	randomcrop224,morpho\_erosion,hflip,invert	80.77%	78.78%	78.3%
30	randomcrop224,morpho\_dilation,hflip,gaussianblur	81.52%	78.82%	78.94%
31	randomcrop224,hflip,invert,gaussianblur	81.64%	79.29%	78.9%
32	randomcrop224,affine,hflip	80.87%	79.23%	79.04%
33	randomcrop224,affine,invert,gaussianblur	80.54%	79.98%	79.8%
34	randomcrop224,morpho\_erosion,colorjitter,gray	80.95%	79.86%	79.98%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
35	randomcrop224,colorjitter,hflip,gaussianblur	80.58%	79.37%	78.36%
36	randomcrop224,colorjitter,invert,gaussianblur	85.26%	79.49%	79.59%
37	randomcrop224,invert,gray	80.98%	78.94%	78.3%
38	randomcrop224,gaussianblur,gray	86.55%	79.45%	79.43%
39	randomcrop224,affine,hflip,gaussianblur	80.32%	79.66%	79.39%
40	randomcrop224,morpho_dilation,hflip	81.64%	79.45%	78.94%
41	randomcrop224,morpho_erosion,colorjitter,invert	81.91%	79.35%	79.39%
42	randomcrop224,morpho_dilation,affine,colorjitter	81.87%	80.62%	80.76%
43	randomcrop224,morpho_erosion,gray	81.97%	79.7%	79.22%
44	randomcrop224,morpho_dilation,gaussianblur	85.03%	79.96%	80.16%
45	randomcrop224,invert,gaussianblur	82.33%	79.9%	78.73%
46	randomcrop224,morpho_erosion,affine,colorjitter	82.63%	80.54%	79.73%
47	randomcrop224,morpho_erosion,hflip,gray	82.3%	78.9%	78.55%
48	randomcrop224,morpho_dilation,affine,gaussianblur	82.04%	80.15%	79.69%
49	randomcrop224,affine	83.99%	80.29%	80.31%
50	randomcrop224,affine,invert,gray	80.51%	79.74%	78.9%
51	randomcrop224,morpho_erosion,invert,gray	84.23%	78.88%	78.92%
52	randomcrop224,affine,gray	77.75%	80.04%	79.3%
53	randomcrop224,hflip,gaussianblur,gray	80.49%	79.06%	78.04%
54	randomcrop224,morpho_erosion,morpho_dilation,gaussianblur	86.10%	80.23%	79.82%
55	randomcrop224,morpho_erosion	85.1%	80.39%	80.04%
56	randomcrop224,morpho_erosion,gaussianblur	87.14%	79.78%	79.3%
57	randomcrop224,morpho_dilation,invert,gray	80.57%	78.96%	78.79%
58	randomcrop224,colorjitter,gaussianblur,gray	80.82%	79.74%	79.77%
59	randomcrop224,morpho_dilation,affine,hflip	78.1%	78.96%	78.18%
60	randomcrop224,morpho_dilation	83.12%	80.17%	79.77%
61	randomcrop224,morpho_erosion,colorjitter,gaussianblur	85.92%	80.05%	79.84%
62	randomcrop224,colorjitter,hflip	82.63%	79.6%	79.51%
63	randomcrop224,morpho_erosion,colorjitter,hflip	81.09%	78.9%	79.1%
64	randomcrop224,morpho_erosion,invert,gaussianblur	86.09%	79.88%	79.67%
65	randomcrop224,colorjitter,invert,gray	79.07%	78.74%	78.12%
66	randomcrop224	82.16%	80.0%	79.78%
67	randomcrop224,morpho_dilation,colorjitter,hflip	81.08%	78.68%	78.18%
68	randomcrop224,morpho_dilation,hflip,invert	80.67%	78.57%	77.83%
69	randomcrop224,morpho_erosion,invert	80.89%	80.07%	80.16%
70	randomcrop224,affine,hflip,gray	79.6%	78.57%	78.26%
71	randomcrop224,morpho_dilation,colorjitter	85.71%	80.31%	79.39%
72	randomcrop224,morpho_erosion,affine,invert	82.56%	80.25%	80.06%
73	randomcrop224,morpho_dilation,affine,gray	80.61%	80.33%	79.88%
74	randomcrop224,colorjitter,invert	82.06%	79.6%	79.71%
75	randomcrop224,morpho_dilation,invert	85.76%	79.94%	80.06%
76	randomcrop224,morpho_erosion,morpho_dilation,gray	81.6%	79.64%	79.98%
77	randomcrop224,affine,colorjitter,hflip	80.69%	79.06%	78.36%
78	randomcrop224,morpho_dilation,invert,gaussianblur	85.07%	80.13%	79.63%
79	randomcrop224,gray	83.08%	80.37%	79.45%
80	randomcrop224,colorjitter,gaussianblur	83.68%	79.92%	80.27%
81	randomcrop224,morpho_erosion,affine,gaussianblur	82.2%	80.78%	80.22%
82	randomcrop224,colorjitter,hflip,gray	80.81%	79.0%	78.75%
83	randomcrop224,morpho_dilation,gray	80.23%	79.78%	79.14%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
84	randomcrop224,hflip,invert,gray	78.96%	78.29%	77.79%
85	randomcrop224,colorjitter,hflip,invert	83.56%	78.82%	78.22%
86	randomcrop224,morpho_erosion,morpho_dilation	84.97%	80.21%	79.47%
87	randomcrop224,morpho_erosion,hflip,gaussianblur	82.35%	79.51%	78.61%
88	randomcrop224,morpho_erosion,gaussianblur,gray	82.06%	80.0%	80.04%
89	randomcrop224,morpho_erosion,affine,gray	81.0%	80.37%	80.23%
90	randomcrop224,morpho_dilation,colorjitter,invert	81.44%	79.8%	79.69%
91	randomcrop224,morpho_erosion,affine,hflip	77.75%	78.86%	78.22%
92	randomcrop224,morpho_dilation,affine,invert	81.75%	80.39%	79.73%
93	randomcrop224,affine,hflip,invert	80.18%	78.51%	78.47%

TABLE A.21: Performance of SimCLR model ResNet-18 architecture across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset with backbone

#### A.6.4 SimCLR model using ResNet-18 Pre-training on Alpub dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop198,morpho_dilation,hflip	99.99%	99.99%	99.99%
2	randomcrop198,colorjitter,hflip,invert	99.97%	99.99%	99.98%
3	randomcrop198,hflip,gray	99.99%	99.99%	99.97%
4	randomcrop198,invert,gaussianblur,gray	99.98%	99.99%	99.99%
5	randomcrop198,morpho_dilation,affine,colorjitter	99.98%	99.99%	99.97%
6	randomcrop198,morpho_dilation,affine	99.99%	99.99%	99.98%
7	randomcrop198,affine,gaussianblur	99.98%	99.99%	99.99%
8	randomcrop198,affine,colorjitter	99.96%	99.97%	99.96%

TABLE A.22: Performance of SimCLR model ResNet-18 architecture across various Data Augmentation combinations pertaining on Alpub Dataset

#### A.6.5 SimCLR Model Using ResNet-18 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset Without Backbone

No.	Transform Type	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_dilation,hflip	19.21%	20.02%	19.16%
2	randomcrop224,invert,gaussianblur,gray	21.66%	22.35%	21.78%
3	randomcrop224,hflip,gray	21.38%	22.29%	21.15%
4	<b>randomcrop224,colorjitter,hflip,invert</b>	<b>26.21%</b>	<b>26.83%</b>	<b>25.13%</b>
5	randomcrop224,morpho_dilation,affine,colorjitter	26.08%	26.54%	25.36%
6	randomcrop224,affine,gaussianblur	18.40%	19.40%	18.00%
7	randomcrop224,morpho_dilation,affine	17.83%	17.99%	17.61%
8	<b>randomcrop224,affine,colorjitter</b>	<b>25.91%</b>	<b>26.76%</b>	<b>25.50%</b>

No	Augmentations	Train Acc	Validation Acc	Test Acc
----	---------------	-----------	----------------	----------

TABLE A.23: Performance of SimCLR model ResNet-18 architecture across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset without backbone

#### A.6.6 SimCLR Model Using ResNet-18 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset With Backbone

No	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_dilation,hflip	78.64%	76.98%	76.75%
<b>2</b>	<b>randomcrop224,colorjitter,hflip,invert</b>	<b>76.95%</b>	<b>77.55%</b>	<b>76.14%</b>
3	randomcrop224,hflip,gray	75.70%	76.32%	75.62%
4	randomcrop224,invert,gaussianblur, gray	78.68%	77.26%	76.81%
5	randomcrop224,morpho_dilation,affine,colorjitter	78.75%	79.08%	78.20%
6	randomcrop224,morpho_dilation,affine	80.02%	79.19%	79.47%
<b>7</b>	<b>randomcrop224,affine,gaussianblur</b>	<b>79.46%</b>	<b>79.55%</b>	<b>78.81%</b>
8	randomcrop224,affine,colorjitter	78.91%	79.53%	79.33%

TABLE A.24: Performance of SimCLR model ResNet-18 architecture across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset with backbone

#### A.6.7 SimCLR model using ResNet-50 Pre-training on ICDAR dataset

No	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop198,hflip	99.97%	100.00%	99.95%
2	randomcrop198,morpho_dilation,gaussianblur	99.96%	99.97%	99.92%
3	randomcrop198,morpho_dilation,colorjitter,hflip	99.91%	99.95%	99.82%
4	randomcrop198,affine,colorjitter,gray	99.76%	99.80%	99.73%
5	randomcrop198,morpho_erosion,morpho_dilation, gaussianblur	99.97%	99.98%	99.95%
6	randomcrop198,morpho_erosion,morpho_dilation,affine	99.87%	99.87%	99.80%
7	randomcrop198,morpho_dilation,invert,gray	99.93%	99.92%	99.86%
8	randomcrop198,hflip,gaussianblur,gray	99.90%	99.96%	99.93%
9	randomcrop198,colorjitter,invert	99.97%	99.96%	99.97%
10	randomcrop198,morpho_dilation,affine,gray	99.86%	99.87%	99.85%
11	randomcrop198,gaussianblur	99.96%	99.98%	99.88%
12	randomcrop198,morpho_erosion,morpho_dilation, colorjitter	99.94%	99.94%	99.87%
13	randomcrop198,morpho_dilation,invert	99.90%	99.97%	99.98%
14	randomcrop198,morpho_dilation,affine,invert	99.74%	99.87%	99.86%
15	randomcrop198,affine,hflip	99.77%	99.87%	99.70%
16	randomcrop198,morpho_dilation,affine	99.84%	99.91%	99.77%
17	randomcrop198,affine	99.83%	99.87%	99.79%
18	randomcrop198,morpho_erosion,affine,colorjitter	99.78%	99.85%	99.75%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
19	randomcrop198,morpho_dilation,gaussianblur,gray	99.95%	99.98%	99.91%
20	randomcrop198,affine,invert	99.83%	99.90%	99.87%
21	randomcrop198,morpho_erosion,affine,invert	99.81%	99.87%	99.86%
22	randomcrop198,colorjitter,invert,gaussianblur	99.92%	99.97%	99.81%
23	randomcrop198,affine,gaussianblur,gray	99.85%	99.87%	99.82%
24	randomcrop198,colorjitter,gaussianblur	99.93%	99.93%	99.92%
25	randomcrop198,colorjitter,hflip,gaussianblur	99.96%	99.98%	99.97%
26	randomcrop198,gaussianblur,gray	99.95%	99.98%	99.91%
27	randomcrop198,hflip,gray	99.93%	99.97%	99.90%
28	randomcrop198,hflip,invert,gaussianblur	99.95%	99.99%	99.88%
29	randomcrop198,invert	99.96%	99.97%	99.99%
30	randomcrop198,hflip,gaussianblur	99.96%	99.98%	99.98%
31	randomcrop198,morpho_dilation,hflip,gray	99.77%	99.95%	99.85%
32	randomcrop198,affine,colorjitter,hflip	99.70%	99.82%	99.75%
33	randomcrop198,morpho_erosion,gaussianblur	99.98%	99.97%	99.94%
34	randomcrop198,morpho_dilation,colorjitter,invert	99.89%	99.92%	99.86%
35	randomcrop198,colorjitter,gray	99.92%	99.93%	99.93%
36	randomcrop198,morpho_dilation,invert,gaussianblur	99.94%	99.96%	99.91%
37	randomcrop198,morpho_dilation,gray	99.97%	99.99%	99.95%
38	randomcrop198,affine,colorjitter	99.76%	99.88%	99.87%
39	randomcrop198,morpho_erosion,morpho_dilation,hflip	99.95%	99.96%	99.91%
40	randomcrop198,affine,colorjitter,invert	99.80%	99.78%	99.77%
41	randomcrop198,hflip,invert,gray	99.93%	99.93%	99.82%
42	randomcrop198,morpho_dilation,colorjitter,gaussianblur	99.95%	99.93%	99.88%
43	randomcrop198,invert,gaussianblur,gray	99.97%	100.00%	99.95%
44	randomcrop198,affine,invert,gray	99.83%	99.87%	99.71%
45	randomcrop198,colorjitter,hflip,gray	99.89%	99.91%	99.79%
46	randomcrop198,morpho_erosion,morpho_dilation,invert	99.95%	99.97%	99.92%
47	randomcrop198,morpho_erosion,colorjitter,gaussianblur	99.96%	99.95%	99.94%
48	randomcrop198	99.93%	99.97%	99.94%
49	randomcrop198,morpho_dilation,hflip,invert	99.87%	99.95%	99.89%
50	randomcrop198,morpho_erosion,affine,gaussianblur	99.85%	99.89%	99.80%
51	randomcrop198,affine,invert,gaussianblur	99.83%	99.86%	99.85%
52	randomcrop198,morpho_erosion,colorjitter,gray	99.88%	99.87%	99.75%
53	randomcrop198,morpho_erosion,hflip,gaussianblur	99.96%	99.98%	99.87%
54	randomcrop198,colorjitter,hflip	99.93%	99.97%	99.90%
55	randomcrop198,colorjitter,gaussianblur,gray	99.88%	99.91%	99.86%
56	randomcrop198,affine,hflip,invert	99.67%	99.88%	99.83%
57	randomcrop198,morpho_erosion,colorjitter,hflip	99.93%	99.91%	99.86%
58	randomcrop198,morpho_erosion,colorjitter,invert	99.86%	99.92%	99.73%
59	randomcrop198,invert,gray	99.96%	99.98%	99.92%
60	randomcrop198,morpho_erosion,invert,gaussianblur	99.94%	99.98%	99.90%
61	randomcrop198,morpho_dilation,colorjitter,gray	99.89%	99.93%	99.87%
62	randomcrop198,morpho_erosion,colorjitter	99.95%	99.92%	99.87%
63	randomcrop198,colorjitter,hflip,invert	99.93%	99.90%	99.91%
64	randomcrop198,morpho_erosion,hflip	99.96%	99.97%	99.88%
65	randomcrop198,hflip,invert	99.95%	99.98%	99.95%
66	randomcrop198,morpho_erosion,morpho_dilation,gray	99.92%	99.97%	99.86%
67	randomcrop198,morpho_dilation,affine,hflip	99.86%	99.87%	99.76%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
68	randomcrop198,morpho_dilation,colorjitter	99.97%	99.96%	99.85%
69	randomcrop198,affine,gaussianblur	99.86%	99.90%	99.75%
70	randomcrop198,morpho_erosion,invert	99.93%	99.97%	99.90%
71	randomcrop198,morpho_dilation,hflip	99.91%	99.95%	99.94%
72	randomcrop198,morpho_erosion,affine,gray	99.75%	99.81%	99.69%
73	randomcrop198,morpho_dilation,hflip,gaussianblur	99.93%	99.94%	99.92%
74	randomcrop198,colorjitter,invert,gray	99.90%	99.92%	99.86%
75	randomcrop198,invert,gaussianblur	99.97%	99.98%	99.94%
76	randomcrop198,morpho_erosion,hflip,gray	99.96%	99.97%	99.90%
77	randomcrop198,morpho_erosion,gray	99.93%	99.97%	99.90%
78	randomcrop198,morpho_erosion,gaussianblur,gray	99.92%	99.96%	99.93%
79	randomcrop198,morpho_erosion,morpho_dilation	99.94%	99.97%	99.92%
80	randomcrop198,morpho_erosion,invert,gray	99.87%	99.95%	99.86%
81	randomcrop198,affine,hflip,gaussianblur	99.81%	99.82%	99.80%
82	randomcrop198,affine,colorjitter,gaussianblur	99.86%	99.82%	99.79%
83	randomcrop198,morpho_dilation,affine,gaussianblur	99.84%	99.89%	99.87%
84	randomcrop198,morpho_erosion,hflip,invert	99.92%	99.97%	99.93%
85	randomcrop198,morpho_dilation	99.93%	99.98%	99.87%
86	randomcrop198,morpho_erosion,affine	99.85%	99.88%	99.77%
87	randomcrop198,affine,gray	99.85%	99.88%	99.79%
88	randomcrop198,colorjitter	99.91%	99.95%	99.90%
89	randomcrop198,gray	99.96%	99.98%	99.89%
90	randomcrop198,morpho_erosion	99.97%	99.97%	99.95%
91	randomcrop198,morpho_erosion,affine,hflip	99.82%	99.84%	99.73%
92	randomcrop198,affine,hflip,gray	99.78%	99.81%	99.69%
93	randomcrop198,morpho_dilation,affine,colorjitter	99.74%	99.80%	99.73%

TABLE A.25: Performance of SimCLR model ResNet-50 architecture across various Data Augmentation combinations pertaining on ICDAR Dataset

#### A.6.8 SimCLR Model Using ResNet-50 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset Without Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
gray	randomcrop224,affine,gaussianblur,gray	28.14%	29.71%	28.79%
1				
2	randomcrop224,hflip,gaussianblur	28.14%	28.6%	27.89%
3	randomcrop224,affine,invert,gaussianblur	29.14%	30.12%	29.55%
4	randomcrop224,gray	30.3%	30.93%	29.39%
5	randomcrop224,morpho_erosion,morpho_dilation,affine	27.24%	27.07%	27.61%
6	randomcrop224,morpho_erosion,hflip,invert	28.41%	29.13%	27.71%
7	randomcrop224,morpho_dilation,colorjitter,gaussianblur	28.39%	29.18%	27.44%
8	randomcrop224,colorjitter,gaussianblur	32.99%	33.45%	33.33%
9	randomcrop224	25.93%	26.6%	24.93%
10	randomcrop224,affine,hflip	22.02%	22.31%	21.49%
11	randomcrop224,hflip,gaussianblur,gray	25.52%	25.84%	24.89%
12	randomcrop224,affine,colorjitter	32.8%	35.15%	33.86%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
13	randomcrop224,gaussianblur,gray	25.17%	25.9%	23.99%
14	randomcrop224,morpho_erosion,morpho_dilation,gray	28.68%	28.28%	27.48%
15	randomcrop224,morpho_dilation,colorjitter,invert	30.83%	31.22%	29.8%
16	randomcrop224,morpho_erosion,hflip,gaussianblur	24.46%	25.13%	24.03%
17	randomcrop224,morpho_erosion,morpho_dilation	24.88%	26.07%	24.64%
18	randomcrop224,morpho_dilation,affine,colorjitter	32.23%	33.59%	33.05%
19	randomcrop224,morpho_dilation,gray	25.04%	26.07%	24.05%
20	randomcrop224,morpho_erosion,affine,gaussianblur	24.04%	24.39%	23.89%
21	randomcrop224,affine,gaussianblur	23.37%	24.41%	23.84%
22	randomcrop224,invert,gray	25.76%	26.54%	24.83%
23	randomcrop224,morpho_erosion,affine,gray	27.54%	28.5%	27.2%
24	randomcrop224,morpho_erosion,colorjitter,hflip	31.21%	31.55%	29.33%
25	randomcrop224,morpho_dilation,hflip,gray	25.21%	26.17%	24.85%
26	randomcrop224,affine,colorjitter,gaussianblur	31.39%	33.39%	32.23%
27	randomcrop224,colorjitter,hflip,gaussianblur	28.03%	27.89%	26.4%
28	randomcrop224,morpho_erosion,affine	25.3%	25.46%	25.62%
29	randomcrop224,morpho_dilation,affine,invert	29.97%	31.32%	31.17%
30	randomcrop224,morpho_dilation,invert,gaussianblur	26.73%	26.99%	26.13%
31	randomcrop224,hflip,gray	25.14%	26.38%	25.05%
32	randomcrop224,morpho_dilation,hflip,invert	26.72%	26.99%	25.69%
33	randomcrop224,morpho_dilation	24.25%	24.47%	23.95%
34	randomcrop224,affine,gray	25.86%	27.38%	26.69%
35	randomcrop224,morpho_erosion,invert,gray	27.3%	27.21%	25.89%
36	randomcrop224,morpho_dilation,affine,hflip	23.29%	24.39%	22.43%
37	randomcrop224,morpho_erosion,morpho_dilation,gaussianblur	24.75%	25.48%	24.07%
38	randomcrop224,morpho_dilation,hflip,gaussianblur	23.22%	24.7%	23.07%
39	randomcrop224,affine,colorjitter,invert	33.18%	33.74%	32.6%
40	randomcrop224,hflip	23.19%	24.15%	22.88%
41	randomcrop224,morpho_dilation,hflip	24.04%	24.06%	23.07%
42	randomcrop224,morpho_erosion,invert,gaussianblur	32.83%	34.04%	32.54%
43	randomcrop224,morpho_erosion,gaussianblur	24.79%	26.03%	24.54%
44	randomcrop224,affine,invert	29.7%	29.5%	29.88%
45	randomcrop224,affine,hflip,invert	29.97%	31.26%	31.06%
46	randomcrop224,morpho_erosion	23.9%	24.02%	23.86%
47	randomcrop224,morpho_erosion,colorjitter,gray	33.48%	34.39%	32.86%
48	randomcrop224,morpho_dilation,invert,gray	29.96%	30.18%	30.57%
49	randomcrop224,affine,colorjitter,hflip	28.78%	29.75%	29.04%
50	randomcrop224,affine,colorjitter,gray	32.32%	33.61%	32.6%
51	randomcrop224,morpho_dilation,colorjitter	31.12%	31.38%	30.16%
52	randomcrop224,hflip,invert	25.93%	26.03%	25.05%
53	randomcrop224,morpho_erosion,colorjitter,invert	40.62%	41.85%	40.49%
54	randomcrop224,morpho_erosion,affine,hflip	23.22%	24.82%	23.54%
55	randomcrop224,morpho_dilation,gaussianblur,gray	30.75%	31.42%	31.27%
56	randomcrop224,colorjitter,invert,gaussianblur	30.71%	30.16%	29.84%
57	randomcrop224,morpho_erosion,invert	26.21%	27.21%	25.79%
58	randomcrop224,morpho_erosion,colorjitter,gaussianblur	32.83%	32.61%	32.8%
59	randomcrop224,morpho_erosion,gray	26.89%	27.32%	25.85%
60	randomcrop224,morpho_dilation,affine,gaussianblur	24.62%	25.33%	25.58%
61	randomcrop224,morpho_dilation,invert	28.72%	29.18%	27.63%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
62	randomcrop224,invert	27.3%	28.07%	26.11%
63	randomcrop224,hflip,invert,gray	26.76%	27.27%	25.66%
64	randomcrop224,colorjitter,gaussianblur,gray	30.64%	31.61%	29.96%
65	randomcrop224,morpho_dilation,affine,gray	27.25%	28.03%	26.95%
66	randomcrop224,morpho_erosion,affine,invert	30.71%	32.67%	31.51%
67	randomcrop224,affine,hflip,gray	25.35%	26.4%	24.99%
68	randomcrop224,colorjitter	28.47%	28.81%	27.65%
69	randomcrop224,morpho_dilation,affine	25.22%	25.9%	26.34%
70	randomcrop224,affine,hflip,gaussianblur	19.88%	20.83%	19.26%
71	randomcrop224,morpho_dilation,colorjitter,gray	30.15%	31.47%	30.47%
72	randomcrop224,morpho_erosion,morpho_dilation,invert	27.08%	27.23%	27.05%
73	randomcrop224,morpho_dilation,colorjitter,hflip	29.73%	30.71%	29.33%
74	randomcrop224,colorjitter,hflip	27.59%	29.24%	26.75%
75	randomcrop224,morpho_erosion,gaussianblur,gray	27.58%	28.24%	27.22%
76	randomcrop224,morpho_erosion,hflip,gray	27.58%	28.13%	27.1%
77	randomcrop224,colorjitter,hflip,invert	29.52%	30.77%	28.73%
78	randomcrop224,colorjitter,invert,gray	28.12%	29.01%	26.75%
79	randomcrop224,morpho_erosion,colorjitter	28.99%	29.52%	28.57%
80	randomcrop224,invert,gaussianblur	27.87%	28.09%	27.79%
81	randomcrop224,morpho_erosion,morpho_dilation,hflip	25.31%	25.43%	25.23%
82	randomcrop224,invert,gaussianblur,gray	25.64%	26.13%	24.72%
83	randomcrop224,morpho_erosion,morpho_dilation,colorjitter	29.3%	29.97%	29.2%
84	randomcrop224,morpho_erosion,affine,colorjitter	32.73%	34.12%	33.35%
85	randomcrop224,gaussianblur	28.13%	27.95%	27.96%
86	randomcrop224,colorjitter,invert	31.11%	30.87%	30.0%
87	randomcrop224,hflip,invert,gaussianblur	28.37%	29.09%	28.63%
88	randomcrop224,morpho_erosion,hflip	22.06%	22.0%	21.94%
89	randomcrop224,affine	22.01%	22.37%	21.76%
90	randomcrop224,colorjitter,hflip,gray	28.46%	28.79%	26.93%
91	randomcrop224,morpho_dilation,gaussianblur	26.06%	26.56%	24.48%
92	randomcrop224,colorjitter,gray	29.27%	28.56%	28.2%
93	randomcrop224,affine,invert,gray	29.7%	29.34%	27.63%

TABLE A.26: Performance of SimCLR model ResNet-50 architecture across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset without backbone

#### A.6.9 SimCLR Model Using ResNet-50 pretrain on ICDAR dataset & fine-tuning on ICDAR Dataset With Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_erosion,morpho_dilation,colorjitter	82.61%	79.37%	79.24%
2	randomcrop224,morpho_dilation,colorjitter,hflip	78.88%	77.98%	76.56%
3	randomcrop224,gaussianblur,gray	80.12%	78.7%	78.69%
4	randomcrop224,morpho_erosion,hflip,gaussianblur	79.74%	77.63%	77.65%
5	randomcrop224,morpho_erosion,colorjitter	83.0%	79.66%	79.12%
6	randomcrop224,morpho_dilation,gray	80.86%	79.37%	78.57%
7	randomcrop224,morpho_dilation,affine,invert	79.62%	78.76%	78.08%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
8	randomcrop224,morpho_dilation,gaussianblur,gray	81.13%	79.45%	79.45%
9	randomcrop224,invert,gaussianblur	82.55%	79.37%	79.94%
10	randomcrop224,morpho_erosion,affine,gaussianblur	78.84%	78.84%	78.9%
11	randomcrop224,morpho_erosion	81.43%	79.21%	79.82%
12	randomcrop224,morpho_dilation,gaussianblur	83.92%	79.84%	79.98%
13	randomcrop224,colorjitter,invert,gaussianblur	82.51%	79.0%	78.88%
14	randomcrop224,colorjitter,gray	82.2%	79.31%	78.24%
15	randomcrop224,morpho_erosion,affine,gray	77.82%	78.82%	78.43%
16	randomcrop224,morpho_dilation,affine,gray	79.28%	79.23%	79.57%
17	randomcrop224,colorjitter,gaussianblur	83.93%	80.05%	80.02%
18	randomcrop224,morpho_erosion,colorjitter,gaussianblur	80.43%	79.76%	78.79%
19	randomcrop224,morpho_erosion,invert,gaussianblur	80.81%	79.41%	78.85%
20	randomcrop224,colorjitter,invert	82.09%	78.96%	79.32%
21	randomcrop224,affine,hflip,invert	77.54%	77.16%	76.52%
22	randomcrop224,colorjitter,hflip,invert	78.03%	77.31%	76.95%
23	randomcrop224,morpho_erosion,invert,gray	80.05%	78.49%	78.12%
24	randomcrop224,morpho_dilation,colorjitter,invert	81.22%	78.55%	78.67%
25	randomcrop224,morpho_erosion,invert	82.51%	79.13%	78.61%
26	randomcrop224,affine,gaussianblur	80.62%	79.35%	78.83%
27	randomcrop224,colorjitter,hflip,gray	78.39%	77.53%	76.69%
28	randomcrop224,morpho_erosion,colorjitter,hflip	79.69%	77.75%	77.61%
29	randomcrop224,hflip,invert	78.87%	77.29%	77.51%
30	randomcrop224,morpho_dilation,colorjitter,gaussianblur	82.93%	79.66%	79.59%
31	randomcrop224,affine,colorjitter	79.31%	79.37%	79.43%
32	randomcrop224,morpho_erosion,morpho_dilation	80.92%	78.8%	78.41%
33	randomcrop224,invert,gray	79.41%	78.37%	78.75%
34	randomcrop224,morpho_erosion,affine,colorjitter	79.28%	79.72%	79.69%
35	randomcrop224,morpho_erosion,hflip,invert	77.51%	76.28%	76.58%
36	randomcrop224,affine,colorjitter,hflip	76.16%	77.61%	77.22%
37	randomcrop224,affine	79.81%	78.65%	78.92%
38	randomcrop224,hflip,gaussianblur	81.4%	78.72%	77.71%
39	randomcrop224,morpho_erosion,colorjitter,invert	83.98%	79.12%	78.79%
40	randomcrop224,colorjitter	80.15%	79.43%	78.49%
41	randomcrop224,colorjitter,hflip	79.27%	78.04%	77.67%
42	randomcrop224,affine,gray	79.2%	79.72%	79.26%
43	randomcrop224,affine,colorjitter,gaussianblur	80.43%	79.17%	79.12%
44	randomcrop224,morpho_erosion,gaussianblur	83.61%	79.59%	79.04%
45	randomcrop224,affine,colorjitter,gray	79.65%	79.29%	78.96%
46	randomcrop224,affine,invert,gray	77.1%	78.08%	79.14%
47	randomcrop224,morpho_erosion,gaussianblur,gray	79.55%	79.51%	79.0%
48	randomcrop224,gaussianblur	85.66%	80.27%	79.63%
49	randomcrop224,morpho_erosion,morpho_dilation,gaussianblur	86.33%	79.19%	78.9%
50	randomcrop224,morpho_dilation,invert,gray	80.94%	79.25%	79.16%
51	randomcrop224,affine,colorjitter,invert	78.85%	78.86%	79.06%
52	randomcrop224,morpho_erosion,morpho_dilation,gray	82.27%	79.57%	77.89%
53	randomcrop224,morpho_dilation,affine,hflip	76.3%	77.16%	76.58%
54	randomcrop224,hflip,gray	77.78%	77.31%	76.42%
55	randomcrop224,morpho_erosion,affine,invert	79.36%	78.55%	78.24%
56	randomcrop224,morpho_dilation,invert,gaussianblur	79.66%	78.94%	78.67%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
57	randomcrop224,hflip,invert,gaussianblur	78.33%	76.9%	76.65%
58	randomcrop224,hflip	78.89%	76.57%	76.95%
59	randomcrop224,morpho_dilation,invert	82.69%	79.04%	79.16%
60	randomcrop224,invert	83.01%	79.13%	79.02%
61	randomcrop224,morpho_erosion,morpho_dilation,hflip	79.75%	77.37%	77.08%
62	randomcrop224,hflip,invert,gray	75.88%	76.49%	76.36%
63	randomcrop224,morpho_erosion,hflip,gray	79.31%	78.35%	77.91%
64	randomcrop224,morpho_erosion,colorjitter,gray	81.6%	79.68%	79.71%
65	randomcrop224,colorjitter,invert,gray	79.36%	77.63%	78.85%
66	randomcrop224,morpho_dilation,colorjitter	82.19%	79.55%	80.08%
67	randomcrop224,affine,hflip	76.12%	77.71%	77.28%
68	randomcrop224,morpho_erosion,morpho_dilation,invert	80.58%	79.39%	79.14%
69	randomcrop224,hflip,gaussianblur,gray	77.9%	77.65%	77.2%
70	randomcrop224,affine,invert	79.44%	78.96%	78.86%
71	randomcrop224,morpho_dilation,hflip	78.19%	77.61%	76.58%
72	randomcrop224,morpho_erosion,affine	79.22%	79.43%	79.53%
73	randomcrop224,morpho_dilation,colorjitter,gray	79.67%	79.15%	78.88%
74	randomcrop224,colorjitter,hflip,gaussianblur	76.41%	77.65%	76.95%
75	randomcrop224,gray	82.43%	79.78%	79.82%
76	randomcrop224,morpho_erosion,morpho_dilation,affine	80.32%	79.7%	79.45%
77	randomcrop224,morpho_erosion,affine,hflip	75.74%	76.75%	76.14%
78	randomcrop224,morpho_dilation,affine,gaussianblur	78.39%	79.13%	78.94%
79	randomcrop224,affine,hflip,gaussianblur	76.81%	77.16%	76.32%
80	randomcrop224,morpho_dilation,affine,colorjitter	80.78%	79.55%	78.98%
81	randomcrop224,morpho_dilation	83.0%	79.6%	79.57%
82	randomcrop224,morpho_dilation,hflip,gaussianblur	79.27%	77.49%	77.06%
83	randomcrop224,invert,gaussianblur,gray	78.34%	78.21%	77.95%
84	randomcrop224,affine,gaussianblur,gray	77.93%	78.82%	78.63%
85	randomcrop224,morpho_erosion,gray	81.41%	79.49%	79.2%
86	randomcrop224,morpho_dilation,hflip,gray	78.06%	77.61%	77.79%
87	randomcrop224,affine,hflip,gray	74.38%	77.24%	76.44%
88	randomcrop224,morpho_erosion,hflip	78.95%	77.65%	77.32%
89	randomcrop224,morpho_dilation,affine	79.48%	79.13%	79.39%
90	randomcrop224,morpho_dilation,hflip,invert	77.54%	77.45%	77.36%
91	randomcrop224,colorjitter,gaussianblur,gray	80.37%	79.53%	79.9%
92	randomcrop224,affine,invert,gaussianblur	79.99%	78.57%	78.88%
93	randomcrop224	84.54%	79.45%	79.49%

TABLE A.27: Performance of SimCLR model ResNet-50 architecture across various Data Augmentation combinations pertaining on ICDAR Dataset & fine-tuning on ICDAR dataset with backbone

#### A.6.10 SimCLR model using ResNet-50 Pre-training on Alpub dataset

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop198,morpho_dilation,hflip	99.98%	99.99%	99.99%
2	randomcrop198,invert,gaussianblur,gray	99.98%	99.99%	99.98%
3	randomcrop198,hflip,gray	99.98%	99.99%	99.97%

No.	Transform Type	Train Acc	Validation Acc	Test Acc
4	randomcrop198,colorjitter,hflip,invert	99.99%	99.99%	99.97%
5	randomcrop198,morpho_dilation,affine,colorjitter	99.97%	99.99%	99.97%
6	randomcrop198,affine,gaussianblur	99.99%	99.99%	99.98%
7	randomcrop198,morpho_dilation,affine	99.99%	99.99%	99.99%
8	randomcrop198,affine,colorjitter	99.98%	99.99%	99.99%

TABLE A.28: Performance of SimCLR model ResNet-150 architecture across various Data Augmentation combinations pertaining on Alpub Dataset

#### A.6.11 SimCLR Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset Without Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_dilation,hflip	17.11%	17.81%	16.87%
2	randomcrop224, hflip, gray	21.26%	21.90%	21.39%
3	randomcrop224,invert,gaussianblur,gray	20.97%	21.61%	20.37%
4	<b>randomcrop224,colorjitter,hflip,invert</b>	<b>24.94%</b>	<b>25.07%</b>	<b>23.95%</b>
5	<b>randomcrop224,morpho_dilation,affine,colorjitter</b>	<b>27.04%</b>	<b>28.50%</b>	<b>27.20%</b>
6	randomcrop224,morpho_dilation,affine	18.87%	19.77%	18.34%
7	randomcrop224,affine,gaussianblur	18.95%	20.32%	19.55%
8	randomcrop224,affine,colorjitter	25.86%	27.17%	26.07%

TABLE A.29: Performance of SimCLR model ResNet-50 architecture across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset without backbone

#### A.6.12 SimCLR Model Using ResNet-50 pretrain on Alpub dataset & fine-tuning on ICDAR Dataset With Backbone

No.	Augmentations	Train Acc	Validation Acc	Test Acc
1	randomcrop224,morpho_dilation,hflip	74.14%	74.87%	74.46%
2	<b>randomcrop224,invert,gaussianblur,gray</b>	<b>74.36%</b>	<b>76.90%</b>	<b>76.59%</b>
3	randomcrop224,hflip,gray	74.33%	74.10%	73.97%
4	randomcrop224,colorjitter,hflip,invert	75.67%	75.71%	74.03%
5	<b>randomcrop224,morpho_dilation,affine,colorjitter</b>	<b>75.69%</b>	<b>78.53%</b>	<b>77.83%</b>
6	randomcrop224,morpho_dilation,affine	74.91%	76.84%	76.09%
7	randomcrop224,affine,gaussianblur	74.77%	76.94%	76.83%
8	randomcrop224,affine,colorjitter	76.21%	77.22%	76.79%

TABLE A.30: Performance of SimCLR model ResNet-50 architecture across various Data Augmentation combinations pertaining on Alpub Dataset & fine-tuning on ICDAR dataset with backbone