

CONTRASTIVE SELF-SUPERVISED LEARNING FOR IMAGE CLASSIFICATION

BY

TAN YONG LE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2021

REPORT STATUS DECLARATION FORM

Title: Contrastive Self-Supervised Learning for Image Classification

Academic Session: Jan 2021

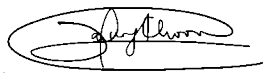
I TAN YONG LE
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.


(Author's signature)

Verified by,


(Supervisor's signature)

Address:

139, Jalan Seri Cempaka 5, Taman

Seri Cempaka, Jalan Junid, 84000

Muar, Johor

Tan Hung Khoon

Supervisor's name

Date: 15/04/2021

Date: 16/04/2021

CONTRASTIVE SELF-SUPERVISED LEARNING FOR IMAGE CLASSIFICATION

BY

TAN YONG LE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2021

DECLARATION OF ORIGINALITY

I declare that this report entitled “**CONTRASTIVE SELF-SUPERVISED LEARNING FOR IMAGE CLASSIFICATION**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature

:



Name

:

Tan Yong Le

Date

:

15/04/2021

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude towards my supervisor of this project, Dr. Tan Hung Khoo, who gave me a chance to work on the area of deep learning. Throughout the project development, he always shared his knowledge and gave me constructive advises. I have learnt a lot in my time of developing this project with him. Next, I would like to thank my parents for their unconditional love and support. Without their encouragement, I would never be who I am today. Lastly, I would like to thank my friends for being with me throughout my hard times in my degree life. Thank you very much.

ABSTRACT

In computer vision, most of the existing state-of-the-art results are dominated by models trained in supervised learning approach, where abundant of labelled data is used for training. However, the labelling of data is costly and limited in some fields. Thus, people have introduced a new paradigm that falls under unsupervised learning – self-supervised learning. Through self-supervised learning, pretraining of the model can be conducted without any human-labelled data and the model can learn from the data itself. The model will pretrain on a pretext task first and the pretext task will ensure the model learn some useful representation for the downstream tasks (e.g., classification, object localization and so on).

One of the top performers in the self-supervised learning paradigm is SimCLR by Chen et al. (2020), in which it achieved 76.5% of top 1 accuracy in ImageNet dataset. Chen et al. (2020) proposed a contrastive self-supervised learning approach, where a pair of samples is produced from one image through different data augmentations and the model will learn while trying to find out each image pair within a training batch. However, they include random cropping as one of their data augmentations, where they allow it to possibly crop out 8% from the original image only. Under such extent of cropping, the model could not learn anything useful of the object, as the region can be a background region or contain too little details of the object.

Thus, this project proposes a novel approach to replace random cropping, where a region proposal algorithm is used to propose regions based on low-level features, such as colour, edges and so on. Thus, the regions produced by the algorithm have a higher chance to consist of an object part, thus promoting better learning. As a result, the pretrained model performs better than the model from SimCLR approach in downstream tasks.

TABLE OF CONTENTS

TITLE PAGE	i
DECLARATION OF ORIGINALITY	i
ACKNOWLEDGEMENTS	ii
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF TABLES	vii
LIST OF ABBREVIATIONS	viii
CHAPTER 1: INTRODUCTION	1
1.1 Project Background and Motivation	1
1.2 Problem Statement	2
1.3 Project Objective and Scope	3
1.4 Project Overview	3
CHAPTER 2: LITERATURE REVIEW	5
2.1 Self-Supervised Learning	5
2.2 SimCLR	8
CHAPTER 3: METHODOLOGY	11
3.1 Self-Supervised Learning Framework	11
3.2 Region Proposals for Cropping	12
CHAPTER 4: EXPERIMENTS	17
4.1 Training Setup	17
4.2 Experiments with Region Proposals Algorithm	19
4.3 Best Settings for Region Proposal Algorithm	23

CHAPTER 5: CONCLUSION	26
BIBLIOGRAPHY	27
APPENDICES	29
A. Data Augmentation Details	29
B. Additional Analysis	31
POSTER	34
PLAGIARISM CHECK RESULT	35
FYP 2 CHECKLIST	39
FINAL YEAR PROJECT WEEKLY REPORT	41

List of Figures

Figure Number	Title	Page
Figure 1.1	An image from STL10 dataset	2
Figure 1.2	Possible crop from random cropping in SimCLR	3
Figure 1.3	Possible crop from random cropping in SimCLR	3
Figure 2.1	Exemplar-CNN	5
Figure 2.2	4-class classification through 90° rotations	6
Figure 2.3	Prediction on relative position patches	7
Figure 2.4	Ambiguity set in 3x3 grid	7
Figure 2.5	Overview of SimCLR	9
Figure 3.1	Overview of project framework	12
Figure 3.2	Regions from NMS algorithm with random score	14
Figure 3.3	Regions from NMS algorithm with edge intensity score	15
Figure 3.4	Regions from NMS algorithm with colour variation score	15
Figure 4.1	Area with more edges or colour variety in an image	21
Figure 4.2	Regions proposed to an image	23
Figure B.1	Class distribution of STL10 unlabelled dataset	31
Figure B.2	Top edged regions	32
Figure B.3	Top regions with more colour variety	322

LIST OF TABLES

Table Number	Title	Page
Table 4.1	Performance of fixed feature extractor for NMS algorithm with different objectiveness score	20
Table 4.2	Performance of fixed feature extractor for with or without NMS algorithm	20
Table 4.3	Performance of fixed feature extractor for random or lowest intersection region pairs	22
Table 4.4	Performance of the models after finetuning to entire network	24
Table 4.5	Performance of the models trained with different batch sizes	25
Table 4.6	Performance of the models trained with different training epochs.	25
Table B.1	Performance of fixed feature extractor for single factor evaluation	33

LIST OF ABBREVIATIONS

<i>CNN</i>	Convolutional Neural Network
<i>NMS</i>	Non-Maximum Suppression
<i>SimCLR</i>	Simple Framework for Contrastive Learning of visual Representation
<i>NT-Xent</i>	Normalised Temperature-scaled Cross Entropy

Chapter 1: Introduction

1.1 Project Background and Motivation

In recent years, many CNN models have shown significant progress in computer vision tasks. Various methods are introduced to train a CNN model and they are mainly categorised by different training approaches, such as supervised and unsupervised learning. Currently, the major approach would be supervised learning, where the training is conducted on a dataset that is properly labelled. However, data labelling is costly and time consuming. In some fields, such as medical field are hard to obtain enough training data and their data are hard to be labelled too. Without sufficient training data, overfitting will occur, and the model will yield low performance in production. To solve such issue, people have suggested to pretrain the model with a large dataset and a similar task first, so the pretrained model can perform better when transferring to the intended task. As a result, the model can converge faster, and the overfitting issue can be minimised. However, pretraining of the model still requires a huge amount of labelled data.

In comparison, unlabelled data are literally available everywhere. Abundant of images and videos are uploaded to the Internet at every moment, but most of them are unlabelled or not properly labelled. To utilise these data, a new paradigm that falls under unsupervised learning has been introduced – self-supervised learning. Through self-supervised learning, no human labelling is required, and the model will be pretrained by conducting a pretext task. The pretext task will be designed in a way to ensure the model can capture important data features through the data itself. There are some projects working on self-supervised learning in the past few years. For example, classification on augmented images, where original images are treated as the label (Dosovitskiy, et al., 2015; Gidaris, et al., 2018; Chen et al., 2020) and image patching to determine the original location of patches (Doersch, et al., 2015; Noroozi & Favaro, 2016).

Among the projects in the self-supervised learning paradigm, one of the top performers is SimCLR by Chen, et al. (2020). Chen, et al. trained the model by comparing the image pairs that are augmented from the same image. As a result, the classifier from SimCLR

project achieved a top-1 accuracy of 76.5% on ImageNet dataset (ILSVRC-2012 by Russakovsky, et al., 2015). This result is a huge improvement from the previous state-of-the-art result in self-supervised paradigm and it is quite competitive to the supervised models too. Thus, this project works on top of SimCLR and provide an improvement on the training approach in SimCLR.

1.2 Problem Statement

SimCLR by Chen, et al. is a framework constructed for contrastive pretraining, in which it will produce pairs of samples and the model has to match up all the samples in a batch into pairs. By learning to match up the images, the model can learn useful features to differentiate images from different classes. However, the model from SimCLR still cannot outperform those supervised models. Some potential issues are limiting the performance of SimCLR model. One of the issues identified is the random cropping technique used in producing image pair in SimCLR. Chen et al. allowed the cropping technique to produce a minimum of 8% from the area of the original image. As shown in Figure 1.2 and Figure 1.3, such extent of cropping is not desirable for contrastive learning as it might produce some background regions as well as some regions that have too little details of the object. In this case, the model could not learn anything related to the object, thus the cropping used in SimCLR hurts the learning process of the model.



Figure 1.1: An image extracted from STL10 dataset.



Figure 1.2 & 1.3: Possible crops from random cropping in SimCLR.

1.3 Project Objective and Scope

The main objective of this project is to improve on the implementation of SimCLR, specifically to tackle the issue of random cropping. This is due to random cropping does not produce conducive regions for contrastive learning. Thus, this project aims to produce some better cropping approaches, so that regions consist of different parts of the object can be produced, as such regions promote contrastive learning.

This project starts with designing the cropping algorithm, where the cropping algorithm pre-processes all the images for pretraining first to produce the regions beforehand. Then, pretraining of a model under different cropping approaches and different settings are conducted. Lastly, transfer learning is conducted to train on classification task and the evaluation of the pretrained model is based on their final accuracy on the classification task.

1.4 Project Overview

The aim of this project is to tackle the issue of random cropping in SimCLR project. A region proposals algorithm is used to extract regions from images, where the regions will be used to crop the original image. The region proposal algorithm will make use of low-level features within the image to propose the regions. Thus, the regions produced are more likely to contain useful information (i.e., object part) for contrastive learning. As a result,

the model pretrained by this approach has successfully learned a better representation and performed better than the model from the original approach in SimCLR project.

The details of this project are presented in this report. This report consists of 5 chapters. Chapter 1 gives an overview of the problem and the project. In chapter 2, various projects in self-supervised paradigm have been reviewed and analysed. In chapter 3, the details of the implementation of the project are presented. Many experiments and analysis are conducted and recorded in Chapter 4. In the last chapter, this project is wrapped up with a conclusion.

Chapter 2: Literature Review

In this chapter, some projects in self-supervised learning paradigm are reviewed. Their strength and limitation are analysed, and this project will adopt their effective techniques and useful findings. Section 2.1 reviews various approaches of self-supervised learning on image data and Section 2.2 reviews SimCLR in depth as this project is mainly based on it.

2.1 Self-Supervised Learning

Self-supervised learning allows people to pretrain a model without any human labelling as it will exploit the label that comes with the data itself. To conduct pretraining through self-supervised learning, a pretext task will be conducted, and the pretext task should lead the model to learn some useful representations (e.g., semantic, or structural information) that are beneficial when the model is used for transfer learning.

The most common approach in self-supervised learning is to use data augmentation on the images and conduct a classification task, where the original image is treated as the label. One of the projects utilising data augmentation is Exemplar-CNN by Dosovitskiy, et al. (2015). Exemplar indicates a thing that is used to be an example of others. Exemplar-CNN treats input images as the exemplar, in which different data augmentation techniques (e.g., contrast, scaling, rotation, etc.) are applied to them to produce distorted images and the distorted images from each exemplar form a surrogate class.

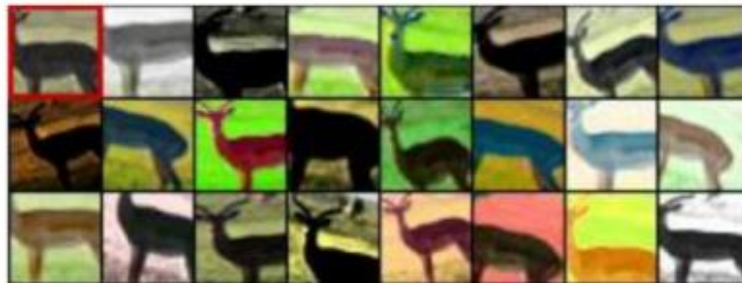


Figure 2.1: Exemplar-CNN – top left image is the original image, and the remaining images are the distorted images from the original one. All these images belong to same surrogate class. (Image source: Dosovitskiy, et al., 2015)

As shown in Figure 2.1, all of the distorted images will form a surrogate class (N input images form N surrogate classes). Then, the model will learn to classify all the images (including the distorted images) into the surrogate class that they belong. However, the issue is that different set of data augmentations applied in training process will vary the result drastically.

In comparison, Gidaris, et al. (2018) has conducted the pretraining the other way round – to predict the augmentation applied to the images. As shown in Figure 2.2, a 4-class classification task is conducted, where the input images are randomly rotated in multiples of 90° (i.e., 0°, 90°, 180° and 270°). As result, the model can learn the relative position of the object parts.

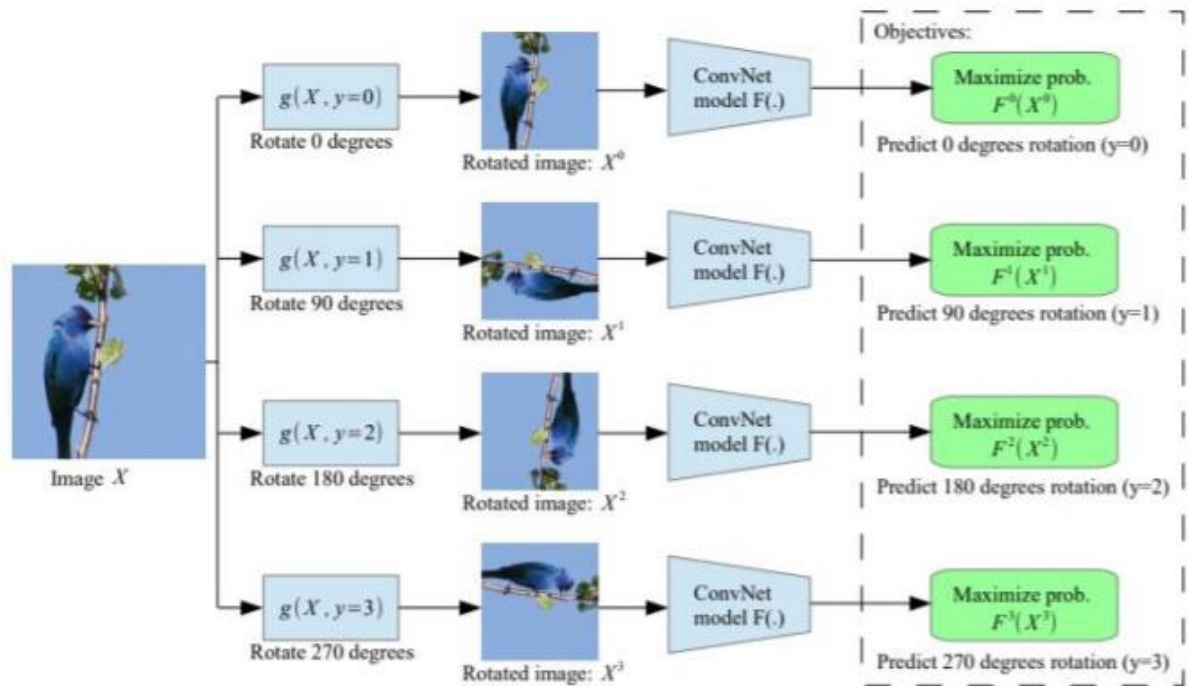


Figure 2.2: Image is rotated into one of the 4 classes (multiples of 90°) and the task is to determine the rotation applied. (Image source: Gidaris, et al., 2018)

On the other hand, Doersch, et al. (2015) have proposed to determine the relative position between the image patch pairs. An image will form a grid of 3x3 patches, where the patches pair is formed by selecting the centre patch and one of the surrounding 8 patches randomly (shown in Figure 2.3).

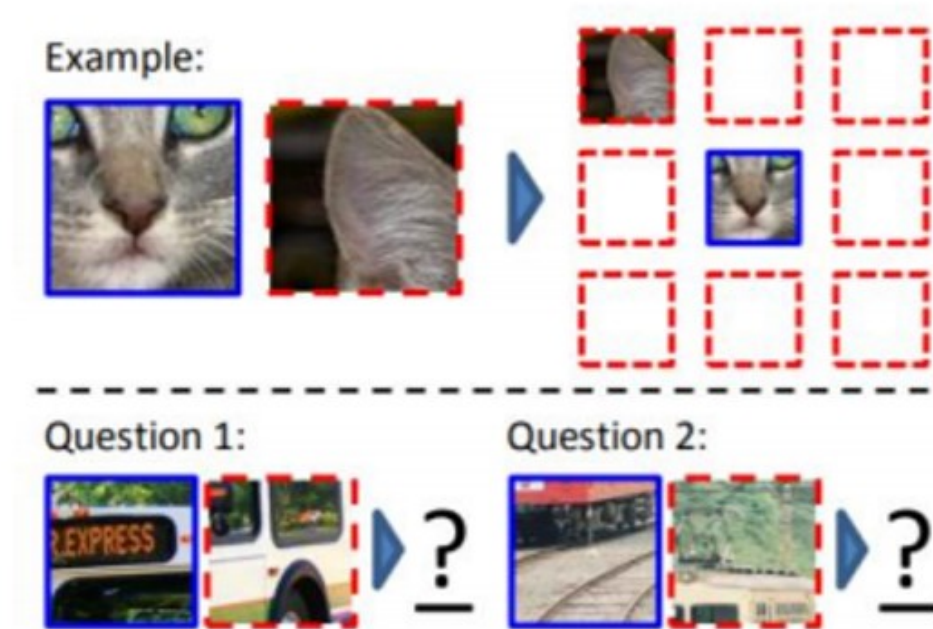


Figure 2.3: The first patch is the centre patch (in blue) and the second patch is selected from the surrounding (in red dotted line). (Image source: Doersch, et al., 2015)

However, the patches pair might introduce some ambiguity during the training. This is because the patches selected can be a background patch or patch that contains too little details of the object.



Figure 2.4: Ambiguity will occur when top-left or top-middle patches is selected to be the second patch in the patches pair. (Image source: Noroozi & Favaro, 2016)

To solve this issue, Noroozi & Favaro (2016) has proposed a Jigsaw puzzle reassembly problem, in which all the patches in 3x3 grid are used during training. The model will learn to arrange the shuffled 3x3 grid to its original state. Through this implementation, the ambiguity is reduced as all the patches in the grid are considered.

2.2 SimCLR

In 2020, Chen, et al. have proposed a simple framework for contrastive learning of visual representation (in short, SimCLR), in which it is a framework to conduct pretraining through contrastive approach. In SimCLR, there are four major components as follow:

- Data augmentation module

This module will produce an image pair for each input image. The image pair is formed by applying two different sets of augmentation. The data augmentation includes cropping, horizontal flipping, colour distortion and Gaussian blurring and they are applied randomly.

- Base encoder

This encoder will extract representation from the augmented images. Any network can be selected as the encoder. By default, ResNet-50 model is used.

- Projection head

Projection head will map the representation extracted by the encoder into latent space. In this case, 2 fully-connected layers are used, and it will map the representation into 128-D latent space.

- Contrastive loss function

Normalised Temperature-Scaled Cross Entropy Loss (NT-Xent) loss is used in SimCLR,

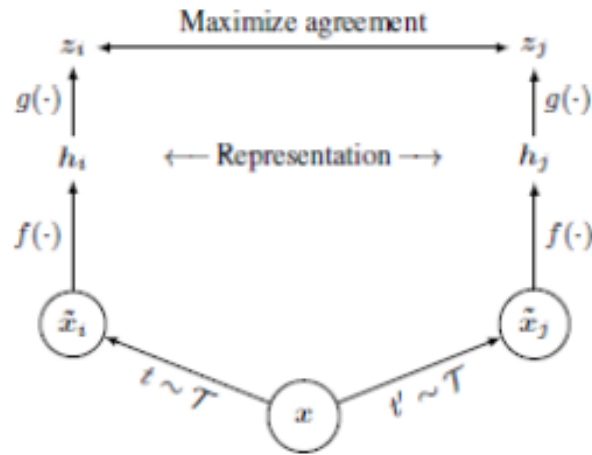


Figure 2.5: x represents the original image, and the augmented pair of x are x_i and x_j . Encoded image, h is then mapped into latent space (as z). (Image source: Chen et al., 2020)

To conduct contrastive pretraining, a mini batch of N samples will be augmented to form a mini batch of $2N$ samples through augmentation. In each mini batch, each sample has a positive sample and $2(N-1)$ negative samples to contrast with. Through NT-Xent loss, the similarity between positive pair of samples will be maximised and the loss against other negative samples are maximised. Thus, the model can discriminate between samples from different classes.

There are few important findings in SimCLR project. Firstly, Chen, et al. has tackled the weakness of Exemplar-CNN, where they have conducted pretraining by using different sets of augmentations. They have concluded that no single augmentation is sufficient for the model to learn a good representation and the best combination of augmentation is cropping with colour distortion. Besides, introducing a projection head between representation and contrastive loss can improve the quality of representation as more useful information can be retained for the downstream tasks. In addition, larger batch size and longer training are preferable as it will benefits a lot to contrastive learning. This is because it will provide more negative samples for the model to contrast with the positive ones.

By combining these findings, SimCLR pretrained model provides better performance over prior state-of-the-art results from self-supervised learning. Although it is quite competitive

to the result from supervised learning, it still cannot outperform them. One of the issues identified is the random cropping used in their data augmentation module. This is because they allow the cropping to produce a minimum area of 8% from the original image size. Thus, random cropping might produce some ambiguity sets during training (as in the project of Doersch, et al., 2015).

Chapter 3: Methodology

This project continues to work on self-supervised learning paradigm. Self-supervised learning is an approach to conduct pretraining without any human labelled data and it can utilise large amount of unlabelled data. In self-supervised learning, the model is trained by conducting a pretext task. However, the final performance of the model on the pretext task is not so important, as the main purpose of the pretext task is to ensure the model to learn a representation that is useful for the downstream tasks (e.g., classification, object localisation and other tasks). Although the purpose of self-supervised learning is to replace conventional pretraining, the performance of the existing pretrained models from self-supervised learning still cannot outperform the pretrained models from supervised learning. This might be due to some issues in designing the pretext task of self-supervised learning.

3.1 Self-Supervised Learning Framework

This project works on top of SimCLR by Chen, et al. (2020). The major framework of SimCLR remains, except the random cropping in data augmentation module. This is because random cropping just produces a region arbitrary without utilising any information from the image. Thus, it will produce some regions that are not useful. Moreover, in their settings, they allow the cropping to produce a region with 8% of the original image area, which is too small. As a result, the regions produced do not contain sufficient details of the object. This causes the representation learned by the model does not contain the targeted semantic or structural meaning of the object, so performing worse in transfer learning. To solve this issue, this project proposes to replace random cropping in SimCLR with region proposal algorithm, where region proposal algorithm makes use of low-level features of object within the image to propose the regions that might contain the object. It turns out that the regions from region proposal algorithm have a higher chance to contain the object if compared to random cropping.

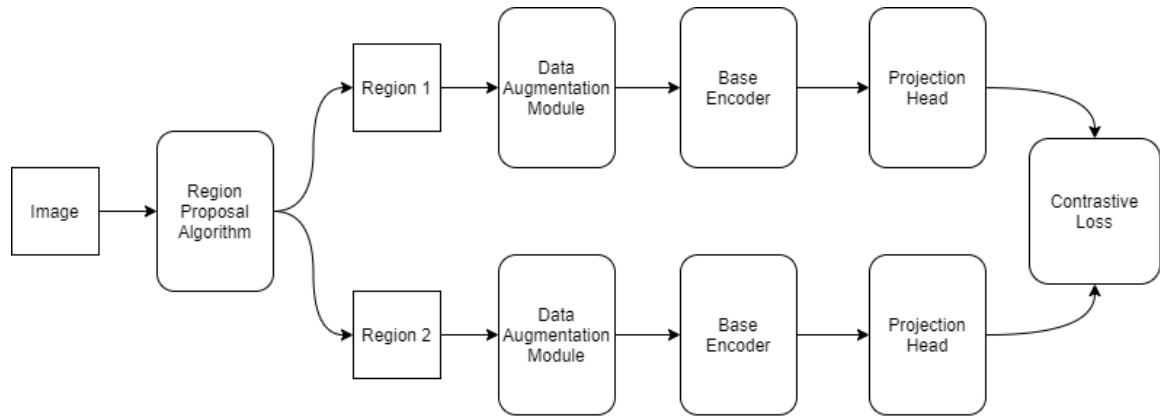


Figure 3.1: Overview of the framework in this project.

As shown in Figure 3.1, a region proposal algorithm is introduced to produce one pair of regions for each input image. The region pairs generated are used to crop the original image and the resulted image pairs are applied with the data augmentation such as horizontal flipping, colour distortion, grey scaling and Gaussian blurring. These data augmentation follows the settings in SimCLR project, and the regions are resized into a fixed shape, as the regions produced comes with different shapes (refer to Appendix A.1 for details). The resulted images are then be encoded and mapped into latent space for contrastive loss. The encoder, projection head and loss function in SimCLR framework remains to be used in this project, which are ResNet-50 model, 2-layer of fully connected layer and NT-Xent loss. Overall, the major modification in this project is to replace the random cropping and the details of the region proposal algorithm are discussed in the following sections.

3.2 Region Proposals for Cropping

There are several region proposal algorithms, such as sliding window (Nakahara, et al., 2017), selective search (Uijlings, et al., 2013), edge boxes (Zitnick & Dollar, 2014), and so on. In this project, selective search algorithm is selected to produce the regions to replace random cropping. Selective search first produces sub segmentation for the input image through over-segmentation approach by Felzenszwalb & Huttenlocher (2004). Then, it uses bottom up approach to combine small regions into larger ones based on their similarity. In this case, selective search algorithm used four different strategies (i.e., colour, texture, size and fill) to compute the similarity between the regions, where the combined result is

the final similarity. As a result, hundreds to thousands of regions will be proposed for each image. These regions are not suitable to be used for cropping yet, as some of them are either too big or too small and most of them are overlapping with each other a lot. Thus, the regions will be further processed and filtered through different mechanisms. The simplified flow of the cropping approach in this project is as below:

- i. Selective search algorithm is used to produce candidate regions from the image.
- ii. In comparison to original area, the regions with too large or small area are removed.
- iii. Regions with high overlapping are removed by non-maximum suppression algorithm.

After the final regions have been generated, they are ready to be used for cropping the original image. For each training epoch, two regions are selected to crop the image to form an image pair for contrastive learning. The details of the cropping and region pair selection are discussed in the following sections.

3.2.1 Filter Mechanisms for Candidate Regions

The region proposal algorithm selected in this project is selective search algorithm (Uijlings, et al., 2013). Quality mode of selective search algorithm is chosen, where it will generate more regions. The size of these regions varies and most of them are overlapping with each other. Therefore, the regions are filtered by their area first, ensuring the regions have sufficient details of the object. The criteria of area filtering are as follow:

- Maximum area: 75% of the original area
- Minimum area: reducing from 50% to 25% (at least produce 50 regions)

After that, non-maximum suppression (NMS) algorithm (Girshick, et al., 2014) is applied to remove the regions with high intersection. In NMS algorithm, the region with the highest objectiveness score is selected as the base region and compared to all other regions. Other regions that exceed overlapping threshold are removed. This process continues with another region that has the next highest objectiveness score until all the regions are used as base region.

- Threshold: increasing from 0.9 to 0.95 (at least produce 10 regions)

- Objectiveness score: random, edge intensity and colour variation

In selective search algorithm, the objectiveness score is not computed. Thus, one of the methods proposed is random selection, where it will randomly choose one of the four coordinates (x1, x2, y1, y2) of the region to be the objectiveness score. Some examples of the regions produced by this method is shown in Figure 3.2.



Figure 3.2: Regions produced by NMS algorithm with random score.

Besides that, an assumption is made that background is less focused when the image is captured, hence producing a smoother texture and less edges will be detected. Thus, the regions with more edges are preferable, so edge intensity is used as the objectiveness score in NMS algorithm. For edge detection, Canny edge detection algorithm (Canny, 1983) is used, where its low threshold and high threshold are fixed at 250 and 500 respectively. Then, the edges detected is divided with the area of the region to compute the edge intensity. The regions produced by NMS algorithm with edge intensity score are shown below:

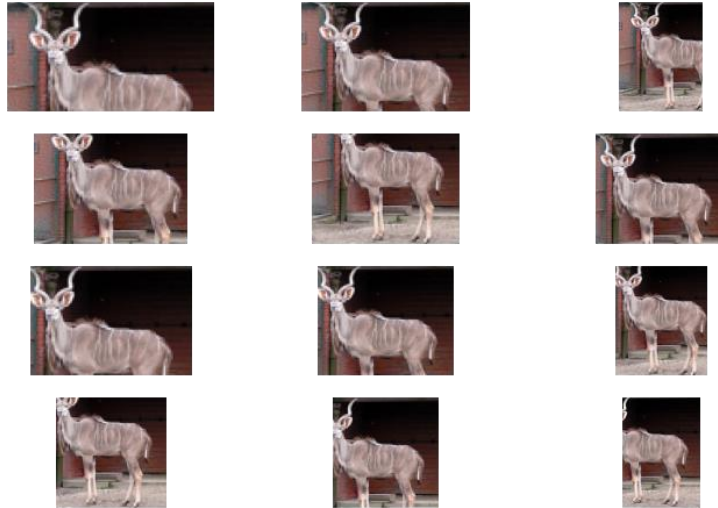


Figure 3.3: Regions produced by NMS algorithm with edge intensity score.

Apart from that, another assumption is made that background of the images should consist of more similar colour, hence there will be more colour variation when the region contains an object part. Thus, the regions with more colour variation are emphasised in this project. In this case, colour variation is used as the objectiveness score for NMS algorithm. To compute the colour variation, the colour values of the pixels within the image are used to calculate the variance, where the larger variance represents more colour variety in the region. Figure 3.4 shows some regions produced through NMS algorithm with colour variation score.

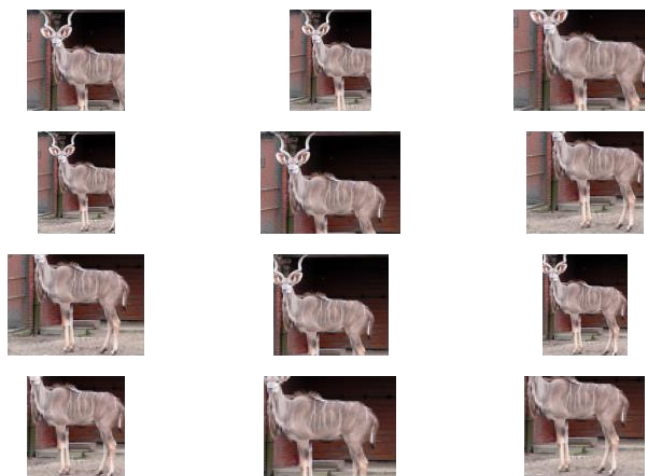


Figure 3.4: Regions produced by NMS algorithm with colour variation score.

3.2.2 Selection Criteria for Region Pairs

As a result, the final regions that most likely contain the object part are extracted and ready to be used for cropping. During the training, for each epoch, two regions have to be selected from the final regions and form a region pair to crop the original image, producing an image pair for contrastive learning. There are two selection mode on how to select the region pair as follow:

- Random: the region pair is randomly sampled from the final regions.
- Least intersection: first region is selected randomly from final regions and second region is chosen by selecting the one with the least intersection to the first region.

Through random selection, more region pairs can be formed during training. Thus, the model can learn a more thorough representation of the object. On the other hand, region pair with least intersection have lesser combination to form during training. However, this can ensure the region pair does not overlap with each other a lot, thus avoiding the model to solve this task through a trivial solution.

CHAPTER 4: Experiments

In this project, contrastive pretraining is conducted under different settings of cropping and region selection. Then, the pretrained model is evaluated on the classification task, where the accuracy represents the quality of the model. In the following sections, different experiments are conducted for different settings presented in Chapter 3 to evaluate the effect of the settings. In the end, the best pretrained model in this project is compared to the original SimCLR method.

4.1 Training Setup

There are two types of training conducted in this project, which are pretraining of the model and finetuning of the model after transfer learning. Thus, in the following sections, two setups for the training are discussed.

4.1.1 Dataset for Pretraining

In this project, a smaller dataset is used for pretraining, which is STL10 unlabelled dataset (Coates, et al., 2011). STL10 unlabelled dataset consists of 100 thousand of 96x96 colour images, where the images are collected from a broader range of classes if compared to STL10 labelled dataset. An analysis is performed on STL10 unlabelled dataset and it shows that the quality of STL10 unlabelled dataset might not be a good choice for pretraining (refer to Appendix B.1). In comparison to SimCLR project, they have used ILSVRC-2012 dataset (Russakovsky, et al., 2015) that consists of 1.2 million of images. Thus, to compare with SimCLR, a pretraining is conducted by using SimCLR approach on STL10 unlabelled dataset.

4.1.2 Pretraining Setup

Due to the limitation of computational resources, this project trains the model for lesser number of iteration and uses a smaller dataset with a smaller batch size for the pretraining. In SimCLR, they pretrained the model for 100 epochs with a default batch size of 4096. However, it is computationally expensive to conduct such pretraining, thus the model is

only pretrained for 50 epochs on a smaller batch size of 32 in this project. Moreover, the dataset chosen for pretraining in this project is STL10 unlabelled dataset, which is a smaller dataset. In short, the default pretraining settings are as follow:

- Training epoch: 50
- Batch size: 32
- Optimiser: Adam (with default settings)
- Linear warmup of 10 epochs.
- Scheduler: CosineAnnealingLR in Pytorch (T_{\max} = number of training batches)

4.1.3 Transfer Learning Setup

Once the models have been pretrained, they are transferred and trained for classification task. The pretrained models act as a feature extractor, where its output layer is removed and replaced with a new output layer that consists of two fully connected layers. This is because for the number of classes for each classification task differs and the last fully connected layer have to output a probability for each class. In transfer learning, there are two ways of training to evaluate the quality of the pretrained models as follow:

- Frozen feature extractor:
Only the output layer will be trained, and it is to examine the quality of the representation learnt by the pretrained model.
- Finetune entire network:
All the layers of the network will be trained, so it is to examine the highest performance that the pretrained model can achieve.

The setups of both training are as follow:

- Training epoch: 50 (with early stopping)
- Batch size: 32
- Optimiser: Adam (with different hyperparameters)
- Scheduler: CosineAnnealingWarmRestarts in Pytorch (T_0 = number of training batches)

For the classification task, the dataset used in this project are CIFAR10 and CIFAR100 (Krizhevsky & Hinton, 2009). The dataset is separated into three sets, which are training set, validation set and test set. The training is conducted to fit on training set, where the model is tested with validation set after each epoch. The model with the best validation loss is saved and the training is stopped once the validation loss is not improved for 5 epochs. The best validation model then perform inference on test set and report the final accuracy. There are few data augmentation used to reduce overfitting, such as weak random cropping (crop between 50% and 100% of original area) and horizontal flipping in training set and resizing and centre cropping for other sets. The details of implementation can refer to Appendix A.2.

4.2 Experiments with Region Proposals Algorithm

The region proposal algorithm selected in this project is selective search algorithm, where it produces many regions, where their size varies, and overlapping with each other. Thus, a further processing is required to find out the better regions. There are different settings presented in Section 3.2. Thus, in the following sections, different experiments are conducted to analyse and choose the best settings for the region proposal algorithm.

4.2.1 Non-Maximum Suppression Algorithm with Selective Search Algorithm

In this project, three different objectiveness scores are used in non-maximum suppression (NMS) algorithm, which are random score, edge intensity and colour variation. NMS algorithm will try to keep the regions with higher objectiveness score, whereby removing other regions that overlap more than the threshold area. Two regions are sampled randomly from the final regions to crop the image for contrastive learning. Three pretrained models are produced based on objectiveness score selection for NMS algorithm. In the experiment, the pretrained models are transferred and frozen to act as a fixed feature extractor, so that the quality of the representation of the pretrained model can be examined. The result of this experiment is shown at below:

Table 4.1: The performance of the fixed feature extractor trained by different objectiveness scores in NMS algorithm.

Model – Objectiveness Score for NMS	CIFAR10 – Accuracy (%)
Random	74.13
Edge Intensity	73.04
Colour Variation	73.57

As shown in Table 4.1, pretrained models when NMS algorithm uses edge intensity and colour variation as objectiveness score performs worse than the one using random objectiveness score. This is because there is not a single factor (i.e., edge or colour) that is sufficient to determine the possibility of containing an object part. Thus, worse regions are generated when only using one factor as the objectiveness score. In comparison to selective search algorithm, it already considers four factors to compute the regions. Thus, the regions proposed by the selective search are good enough, so a random objectiveness score would be sufficient in this case.

4.2.2 Use of Non-Maximum Suppression Algorithm

In this section, an experiment is conducted to examine if non-maximum suppression (NMS) algorithm is necessary to remove the regions with high overlapping. In this case, regions generated after filtering their sizes are used for cropping the original images directly. The pretrained model is also frozen to act as a fixed feature extractor and the result of this experiment is shown below:

Table 4.2: The performance of the fixed feature extractor trained on regions with or without NMS algorithm.

Model – with or without NMS	CIFAR10 – Accuracy (%)
With NMS	74.13
Without NMS	74.81

As shown in Table 4.2, the accuracy produced by the model trained from the regions without NMS algorithm performs better. Without NMS algorithm, most of the regions generated overlap with each other. However, this might not be an issue in this case, as it promotes some deviations from a particular region, thus introducing more randomness for training. In this case, NMS algorithm does not produce regions for better contrastive learning. This is because the number of regions produced by selective search algorithm is not many as the image size in STL10 dataset is quite small. As a result, the number of regions after the regions are removed by NMS algorithm are not sufficient for the training, hence limiting the learning of the model. Thus, NMS algorithm is more necessary when the dataset consists of larger images, where more regions will be produced, so NMS algorithm is required to reduce the number of regions.

4.2.3 Single Factor as Selection Criteria

Some additional experiments are also conducted on the regions without going through NMS algorithm, where a selection criterion is used to select the top regions. In this case, the regions are ranked based on their edge intensity or colour variation, and only the top 50% regions are used for training. However, the result produced by the pretrained models under such settings are worse (refer to Appendix B.2). This is because most of the regions with the top selection criteria are pointing to same area that has more edges or colour variation.

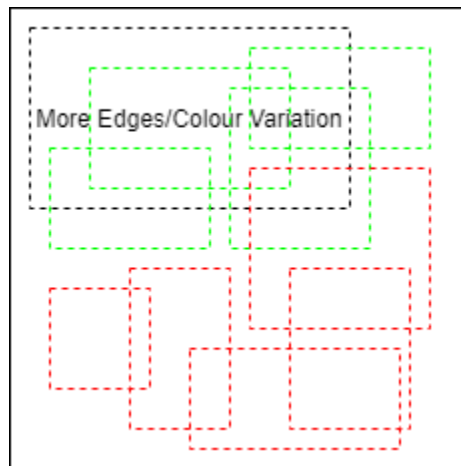


Figure 4.1: Example of an area with more edges or colour variation.

As shown in Figure 4.1, the green regions that are near to more edges or colour variation area are mostly likely to be kept for training, while the red regions are most likely to be removed. Thus, the model is limited to learn as only a particular area of the image is pointed by the regions, hence it cannot learn a thorough representation of the object.

4.2.4 Random Selection for Region Pair

For each training iteration, two images need to be produced by selecting two regions to crop the original image. In this section, experiments are conducted on the different selection methods, which are random selection and lowest intersection selection. In random selection, two regions are selected randomly from the final regions, whereby for lowest intersection selection, one region is selected randomly first and another region is selected by searching other regions for the one with the lowest intersection. The result of the experiments is shown below:

Table 4.3: The performance of the fixed feature extractor trained on random region pairs or lowest intersection region pairs.

Model – with/without NMS	Model – random or lowest intersection region pair	CIFAR10 – Accuracy (%)
Without NMS	Random	74.81
	Lowest Intersection	71.36
With NMS	Random	74.13
	Lowest Intersection	73.93

As shown in Table 4.3, the models trained from region pairs with random selection performs better than the one with least intersection region pair. This can be observed clearly in the case of model without NMS. This is because there are some regions keep being selected during the training, especially for a small or corner region, as they always have lesser intersection with other regions.

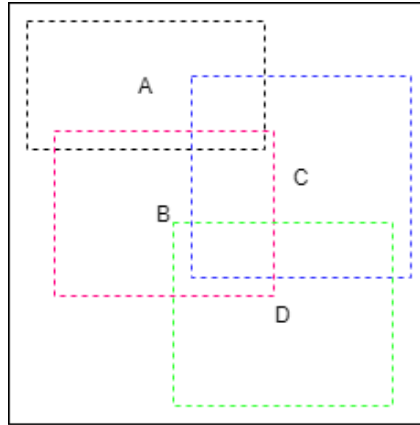


Figure 4.2: Example of regions proposed to an image.

As shown in Figure 4.2, if there is a small region that is located near a corner of the image (e.g., Region A) is proposed, the region will always be selected to be another region for least intersection region pair. Thus, the learning of the model is limited in this case. Besides that, the pairing of the regions for lowest intersection is fixed, thus the model is easier to train with, as there are lesser region pairs to be formed during training. As a result, the pretext task becomes easier to be trained with and the model cannot learn well.

4.3 Best Settings for Region Proposal Algorithm

In Section 4.2, the best performance by the fixed feature extractor is given by the model trained without NMS algorithm and with random region pair. On the other hand, another model is also selected from the one that uses NMS algorithm. Training is conducted on the three models from different objectiveness score in NMS algorithm and the best model among them is the model from random objectiveness score. It can also be observed that random region pair is better than lowest intersection region pair. Thus, in this section, training to entire network is conducted to the following models:

- **Model 1:** trained on random region pairs and the regions are without NMS algorithm.
- **Model 2:** trained on random region pairs and the regions are with NMS algorithm that uses random score.

Besides that, a model is also pretrained by using the original approach in SimCLR project. This model will act as the benchmark for the improvement in this project. Apart from that, another model with default pretraining in PyTorch is also included, where this model is pretrained on ImageNet dataset through supervised approach. Thus, this model will act as the benchmark for the supervised model.

4.3.1 Evaluation on CIFAR10 and CIFAR100

To push the models to achieve the highest result, hyperparameters tuning is conducted. For each set of hyperparameters, the training is conducted for three time and the average accuracy is recorded. The highest accuracy for each model is reported as follow:

Table 4.4: Accuracy of all the models on CIFAR10 and CIFAR100.

Model	CIFAR10 – Accuracy (%)	CIFAR100 – Accuracy (%)
Model 1	86.39	58.45
Model 2	86.24	58.02
SimCLR	85.48	54.57
Supervised	94.44	78.00

As shown in Table 4.4, Model 1 which is trained without NMS performs the best among the self-supervised models. However, Model 1 still cannot outperform the supervised. There are several reasons for the performance gap. First, larger batch size and longer training epoch are required for better contrastive learning. This is because it can provide more negative samples for the model to contrast with. Second, the number of training data used for pretraining in this project is lesser. The supervised model in Table 4.4 is pretrained on ImageNet dataset that consists of 1.2 million of images, while this project only uses STL10 unlabelled dataset that contains 100 thousand of images.

4.3.2 Larger Batch Size and Longer Training for Contrastive Learning

An experiment is conducted to examine if larger batch size and longer training helps contrastive learning. In this experiment, the model is pretrained in a batch size of 128, and the regions are without NMS algorithm and selected randomly when forming the region pair (i.e., Model 1). The result of the model that is trained for 50 epochs is as follow:

Table 4.5: Accuracy of models of different batch sizes.

Model	CIFAR10 – Accuracy (%)	CIFAR100 – Accuracy (%)
Model 1 – Batch size: 32	86.39	58.45
Model 1 – Batch size: 128	87.15 (+0.76)	58.88 (+0.43)

As shown in Table 4.5, the pretrained model from larger batch size performs better under same training epochs. It proves that larger batch size benefits contrastive learning. However, the improvement of the performance is not so significant. This might be due to the difference between the batch sizes is small. Due to the limitation of the computational resources, larger batch size (more than 128) cannot be done. Thus, another experiment is conducted to examine the effect of longer training, where the model is further trained for 25 epochs. The result of this model is shown below:

Table 4.6: Accuracy of model with different training epochs.

Model	CIFAR10 – Accuracy (%)	CIFAR100 – Accuracy (%)
Model 1 – 50 epochs	87.15	58.88
Model 1 – 75 epochs	88.13 (+0.98)	60.97 (+2.09)

As shown in Table 4.6, the improvement by longer training is significant, where the accuracy is improved more than the effect of batch size (from 32 to 128). This might be due to the small difference of batch sizes used for comparison or larger batch size require longer training to get better result.

Chapter 5: Conclusion

In conclusion, this project introduces a novel approach that uses a region proposal algorithm to replace random cropping in SimCLR project. A region proposal algorithm is better than the random cropping in SimCLR project, as it makes use of the low-level features within the image to propose regions. Thus, the regions will have a higher chance to contain an object part and these regions are more desirable, as they can ensure the model to learn some useful representation of the object. As a result, the performance of the pretrained model for transfer learning is better than the model from original SimCLR approach.

In this project, the current best model is trained on the random region pairs, where the regions are without NMS algorithm. The largest training delivered in this project is a training conducted for 75 epochs with a batch size of 128. As a result, this model produces an accuracy of 88.13% and 60.97% for CIFAR10 and CIFAR100 dataset respectively, whereby the default pretrained model in PyTorch produces a result of 94.44% and 78.00% for both datasets respectively.

It is believed that the performance gap can be reduced if the even larger training is conducted. From the result in Section 4.3.2, it shows that the results presented in this project can be further improved by using a larger batch size and longer training. Besides that, the result should be improved significantly when ImageNet dataset is used for pretraining. Thus, the future work of this project is to conduct pretraining on ImageNet with larger batch size and longer training.

BIBLIOGRAPHY

Anon., n.d. *Torchvision Models*. [Online]

Available at: <https://pytorch.org/vision/stable/models.html> [Accessed 16 4 2020].

Canny, J., 1983. A Variational Approach to Edge Detection. *AAAI*, Volume 1983, pp. 54-58.

Chen, T., Kornblith, S., Norouzi, M. & Hinton, G., 2020. A simple framework for contrastive learning of visual representations. *International conference on machine learning*, pp. 1597-1607.

Coates, A., Lee, H. & Ng, A. Y., 2011. An analysis of single-layer networks in unsupervised feature learning. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215-223.

Doersch, C., Gupta, A. & Efros, A. A., 2015. Unsupervised Visual Representation Learning by Context Prediction. *Proceedings of the IEEE international conference on computer vision*, pp. 1422-1430.

Dosovitskiy, A. et al., 2015. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9), pp. 1734-1747.

Felzenszwalb, P. F. & Huttenlocher, D. P., 2004. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2), pp. 167-181.

- Gidaris, S., Singh, P. & Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations.
- Girshick, R., Donahue, J., Darrell, T. & Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580-587.
- Krizhevsky, A. & Hinton, G., 2009. Learning multiple layers of features from tiny images.
- Nakahara, H., Yonekawa, H. & Sato, S., 2017. An object detector based on multiscale sliding window search using a fully pipelined binarized CNN on an FPGA. *2017 international conference on field programmable technology (ICFPT)*, pp. 168-175.
- Noroozi, M. & Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. *European conference on computer vision*, pp. 69-84.
- Russakovsky, O. et al., 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), pp. 211-252.
- Uijlings, J. R. R., Sande, K. E. A. v. d., Gevers, T. & Smeulders, A. W. M., 2013. Selective Search for Object Recognition. *International journal of computer vision*, 104(2), pp. 154-171.
- Zitnick, C. L. & Dollar, P., 2014. Edge boxes: Locating object proposals from edges. *European conference on computer vision*, pp. 391-405.

APPENDICES

A. Data Augmentation Details

A.1 Data Augmentation for Pretraining

For the default pretraining settings, the data augmentation used includes resizing, random flipping, random colour distortion, random grey scaling, and random Gaussian blurring. These augmentations are applied after the images are cropped by the region pairs. The details of the implementation of the data augmentation are shown in PyTorch and OpenCV code as follow:

```
# input_shape represents the original size of the image
transforms.Compose([
    transforms.Resize((input_shape[0], input_shape[0])),
    transforms.RandomHorizontalFlip(),
    transforms.RandomApply([transforms.ColorJitter(
        0.8, 0.8, 0.8, 0.2)], p=0.8),
    transforms.RandomGrayscale(p=0.2),
    GaussianBlur(kernel_size=int(0.1 * input_shape[0])),
    transforms.ToTensor()
])
# cv2 implementation of Gaussian Blurring
class GaussianBlur(object):
    def __init__(self, kernel_size, min=0.1, max=2.0):
        self.min = min
        self.max = max
        self.kernel_size = kernel_size
    def __call__(self, sample):
        sample = np.array(sample)
        prob = np.random.random_sample() # [0,1)
        if prob < 0.5:
            sigma = (self.max - self.min) * np.random.random_sample() +
self.min
            sample = cv2.GaussianBlur(sample, (self.kernel_size,
self.kernel_size), sigma)
        return sample
```


A.2 Data Augmentation for Transfer Learning

For transfer learning, the data augmentations are applied to avoid early overfitting. The data augmentations used are shown as below:

For training set,

```
train_transform = transforms.Compose([
    transforms.RandomResizedCrop(92, (0.5, 1)), # Note this
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor()
])
```

For validation and test set,

```
test_transform = transforms.Compose([
    transforms.Resize(108),
    transforms.CenterCrop(92),
    transforms.ToTensor()
])
```

B. Additional Analysis

B.1 Analysis on STL10 Unlabelled Dataset

The purpose of this analysis is to examine the quality of STL10 unlabelled dataset and its suitability for contrastive learning. First, analysis on the class distribution of the images is conducted. STL10 unlabelled dataset consists of 100 thousand images that are acquired from ImageNet with a broader range of classes, if compared to STL10 labelled dataset. To perform this analysis, an ImageNet-pretrained ResNet-50 model is used to categorise the images. This model has been trained on ImageNet dataset that has 1000 classes and it can classify ImageNet dataset with an accuracy of 76.13% (Anon., n.d.). As a result, this model has classified STL10 unlabelled dataset into 948 classes and the class distribution is shown below:

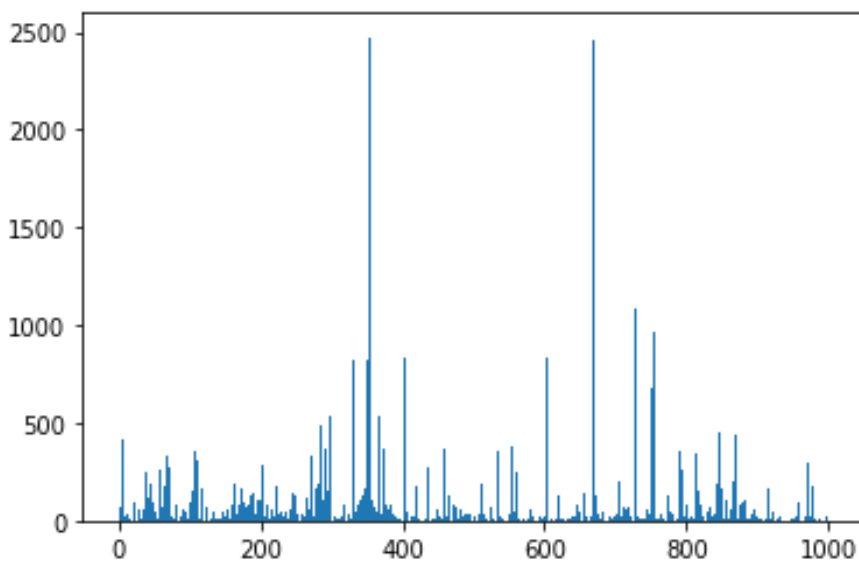


Figure B1: Class distribution of STL10 unlabelled dataset.

As shown in Figure B1, STL10 unlabelled dataset is class imbalanced, where its top-5 frequency classes occupied 10.76% from the entire dataset. This is not so desirable in contrastive learning, as images from same classes will have higher chance to appear in the same training batch, so the model will try to categorise images from same classes into

different classes. This is not desirable as this defeats the purpose of contrastive learning, where the model should learn to differentiate images from different classes.

B.2 Single Factor as Selection Criteria of Top Regions

There are two factors used for this experiment, which are edge intensity and colour variation. The top 50% regions with the higher factor are remained and used to crop the image for contrastive learning. Examples for such regions are shown below:



Figure B2: Regions extracted based on their edge intensity.



Figure B3: Regions extracted based on their colour variation.

As shown in Figure B2 and Figure B3, the regions are mostly pointing to a particular area, thus the model cannot learn a thorough representation of the object, as there are no different views of objects shown to the model to learn. As a result, the performance of the models are the worst among all the settings and their result as the fixed feature extractor are shown below:

Table B1: Performance of fixed feature extractor trained from regions extracted from single factor evaluation.

Model	CIFAR10 – Accuracy (%)
Edge Intensity	70.60
Colour Variation	72.42

POSTER



Faculty of Information and Communication Technology

Contrastive Self Supervised Learning for Image Classification

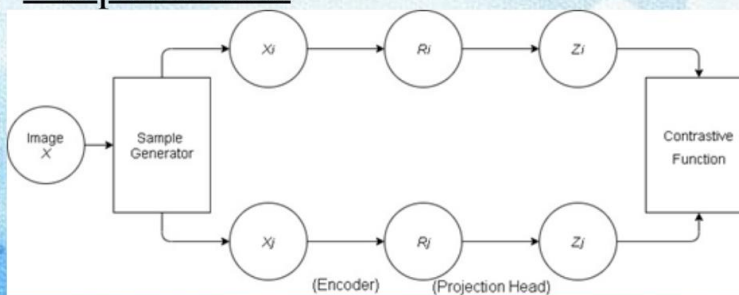
Why self-supervised learning?



When people want to train a neural network, they usually go for supervised training, which requires a labelled dataset. However, labelling is time consuming.

Self-Supervised Learning is introduced as it only requires unlabeled data for training. The concept is to conduct a pretext task to pretrain a model. The pretext task will ensure the model to capture the important data features.

Our pretext task:



Note that:
Original image is the "class label" for the two augmented images.

In a batch, every image will result in two augmented images through different set of augmentations (e.g. cropping). So, the model is **trained** to find out another augmented image from the batch when one augmented image is given.

PLAGIARISM CHECK RESULT

Contrastive Self-Supervised Learning for Image Classification

ORIGINALITY REPORT

2%

SIMILARITY INDEX

1%

INTERNET SOURCES

2%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

tel.archives-ouvertes.fr

Internet Source

1%

2

Changjae Oh, Bumsuh Ham, Hansung Kim, Adrian Hilton, Kwanghoon Sohn. "OCEAN: Object-Centric Arranging Network for Self-supervised Visual Representations Learning", Expert Systems with Applications, 2019

Publication

<1%

3

arxiv.org

Internet Source

<1%

4

"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018

Publication

<1%

5

dspace.library.uvic.ca:8080

Internet Source

<1%

6

"Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020

Publication

<1%

7

Lisa Knopp, Marc Wieland, Michaela Rättich, Sandro Martinis. "A Deep Learning Approach

<1%

for Burned Area Segmentation with Sentinel-2 Data", Remote Sensing, 2020

Publication

- | | | |
|----|---|------|
| 8 | eprints.soton.ac.uk
Internet Source | <1 % |
| 9 | Pritam Sarkar, Ali Etemad. "Self-Supervised Learning for ECG-Based Emotion Recognition", ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020
Publication | <1 % |
| 10 | Qian Xiang, Xiaodan Wang, Rui Li, Guoling Zhang, Jie Lai, Qingshuang Hu. "Fruit Image Classification Based on MobileNetV2 with Transfer Learning Technique", Proceedings of the 3rd International Conference on Computer Science and Application Engineering - CSAE 2019, 2019
Publication | <1 % |
| 11 | Talia Konkle, George A. Alvarez. "Instance-level contrastive learning yields human brain-like representation without category-supervision", Cold Spring Harbor Laboratory, 2020
Publication | <1 % |

Chapter 1: Introduction

1.1 Project Background and Motivation

In recent years, many CNN models have shown significant progress in computer vision tasks. Various methods are introduced to train a CNN model and they are mainly categorised by different training approaches, such as supervised and unsupervised learning. Currently, the major approach would be supervised learning, where the training is conducted on a dataset that is properly labelled. However, data labelling is costly and time consuming. In some fields, such as medical field are hard to obtain enough training data and their data are hard to be labelled too. Without sufficient training data, overfitting will occur, and the model will yield low performance in production. To solve such issue, people

Match Overview

2%

1	tel.archives-ouvertes.fr Internet Source	1%
2	Changjae Oh, Bumsub ... Publication	<1%
3	andiv.org Internet Source	<1%
4	"Computer Vision - EC... Publication	<1%
5	dspace.library.uvic.ca/8... Internet Source	<1%
6	"Computer Vision - EC... Publication	<1%
7	Lisa Knopp, Marc Wiela... Publication	<1%
8	eprints.soton.ac.uk Internet Source	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION
TECHNOLOGY**

Full Name(s) of Candidate(s)	Tan Yong Le
ID Number(s)	17ACB01800
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	Contrastive Self-Supervised Learning for Image Classification

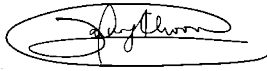
Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u> 2 </u> % Similarity by source Internet Sources: <u> 1 </u> % Publications: <u> 2 </u> % Student Papers: <u> 0 </u> %	
Number of individual sources listed of more than 3% similarity: <u> 0 </u>	

Parameters of originality required and limits approved by UTAR are as Follows:

- (i) Overall similarity index is 20% and below, and**
- (ii) Matching of individual sources listed must be less than 3% each, and**
- (iii) Matching texts in continuous block must not exceed 8 words**

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.



Signature of Supervisor

Name: Tan Hung Khoon

Date: 16/04/2021

Signature of Co-Supervisor

Name: _____

Date: _____

FYP 2 CHECKLIST



UNIVERSITI TUNKU ABDUL RAHMAN

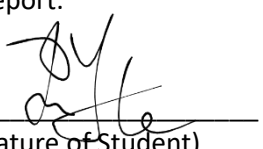
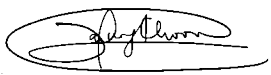
FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB01800
Student Name	Tan Yong Le
Supervisor Name	Ts Dr.Tan Hung Khoon

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

<p>I, the author, have checked and confirmed all the items listed in the table are included in my report.</p> <p> _____ (Signature of Student) Date: 16/04/2021</p>	<p>Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.</p> <p> _____ (Signature of Supervisor) Date: 16/40/2021</p>
--	---

FINAL YEAR PROJECT WEEKLY REPORT

Trimester, Year: Year 3 Sem 3	Study week no.: 2
Student Name & ID: Tan Yong Le 1701800	
Supervisor: Ts Dr. Tan Hung Khoon	
Project Title: Contrastive Self-Supervised Learning with Image Classification	

1. WORK DONE

Implementation of Selective Search algorithm and related pretraining and transfer learning are conducted.

2. WORK TO BE DONE

NMS algorithm and related pretraining

3. PROBLEMS ENCOUNTERED

Slow GPU in Google Colab, thus training takes a long time.

4. SELF EVALUATION OF THE PROGRESS

Need to figure out how to cope with the GPU.

Supervisor's signature

Student's signature

Trimester, Year: Year 3 Sem 3	Study week no.: 4
Student Name & ID: Tan Yong Le 1701800	
Supervisor: Ts Dr.Tan Hung Khoon	
Project Title: Contrastive Self-Supervised Learning with Image Classification	

1. WORK DONE Implementation of NMS algorithm for different objectiveness score and related pretraining and finetuning are conducted.
2. WORK TO BE DONE Region pair selection for pretraining
3. PROBLEMS ENCOUNTERED Slow GPU in Google Colab, thus training takes a long time.
4. SELF EVALUATION OF THE PROGRESS Need to figure out how to cope with the GPU.

Supervisor's signature

Student's signature

Trimester, Year: Year 3 Sem 3	Study week no.: 6
Student Name & ID: Tan Yong Le 1701800	
Supervisor: Ts Dr.Tan Hung Khoon	
Project Title: Contrastive Self-Supervised Learning with Image Classification	

1. WORK DONE Implementation of region pair selection with random selection or least intersection selection
2. WORK TO BE DONE Analysis on different settings
3. PROBLEMS ENCOUNTERED Slow GPU in Google Colab, thus training takes a long time.
4. SELF EVALUATION OF THE PROGRESS Open different Google account to cope with the Colab sessions.

Supervisor's signature

Student's signature

Trimester, Year: Year 3 Sem 3	Study week no.: 8
Student Name & ID: Tan Yong Le 1701800	
Supervisor: Ts Dr.Tan Hung Khoon	
Project Title: Contrastive Self-Supervised Learning with Image Classification	

1. WORK DONE Analysis on different settings are conducted and the best settings is found.
2. WORK TO BE DONE Try to improve further the performance of the pretrained model through hyperparameter tuning.
3. PROBLEMS ENCOUNTERED Hyperparameter tuning takes a long time to find out a good set of hyperparameters.
4. SELF EVALUATION OF THE PROGRESS Need to learn some techniques of hyperparameter tuning.

Supervisor's signature

Student's signature

Trimester, Year: Year 3 Sem 3	Study week no.: 10
Student Name & ID: Tan Yong Le 1701800	
Supervisor: Ts Dr.Tan Hung Khoon	
Project Title: Contrastive Self-Supervised Learning with Image Classification	

1. WORK DONE Hyperparameters tuning of training for transfer learning.
2. WORK TO BE DONE Continue with hyperparameters tuning as well as compiling the report.
3. PROBLEMS ENCOUNTERED Slow GPU in Google Colab, thus training takes a long time.
4. SELF EVALUATION OF THE PROGRESS -

Supervisor's signature

Student's signature

Trimester, Year: Year 3 Sem 3	Study week no.: 12
Student Name & ID: Tan Yong Le 1701800	
Supervisor: Ts Dr.Tan Hung Khoon	
Project Title: Contrastive Self-Supervised Learning with Image Classification	

1. WORK DONE Important analysis for report and some chapters of the report.
2. WORK TO BE DONE Remaining chapters of the report (as well as some analysis required)
3. PROBLEMS ENCOUNTERED Bad in writing the report.
4. SELF EVALUATION OF THE PROGRESS Need to improve my writing skills.

Supervisor's signature

Student's signature