**Project**

**Permuted Bootstrap You Own Latent** - **PBYOL**

**Or Livne - 203972922**

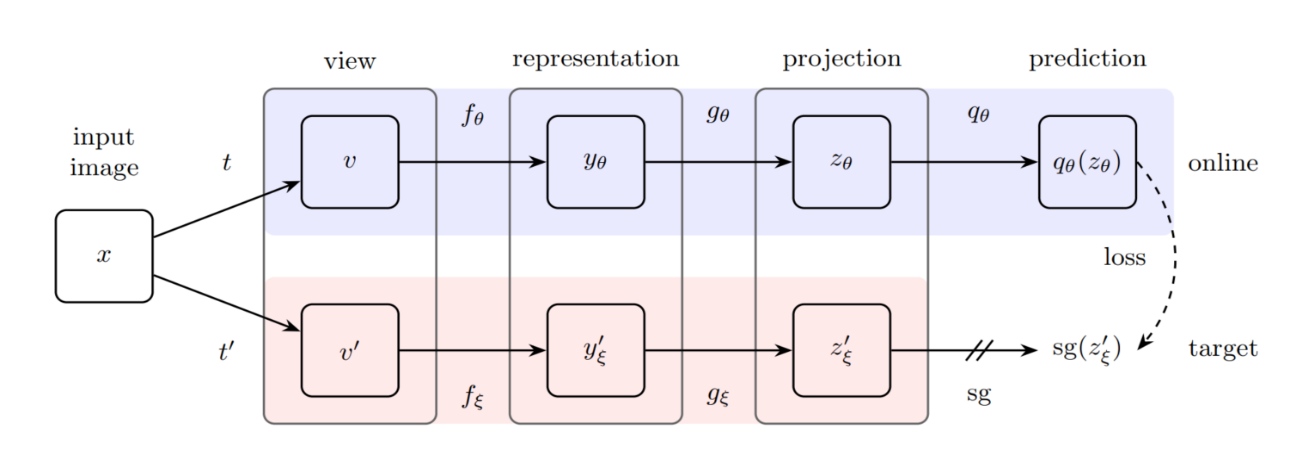
**Abstract:**

* We introduced Permuted Bootstrap Your Own Latent (PBYOL), a new approach to self-supervised vision representation learning.
* PBYOL was inspired by Bootstrap Your Own Latent (BYOL), which is a method for image representation learning.
* The BYOL architecture consists of two neural networks, known as the online and targets networks.
* The optimization of the BYOL model is achieved by training the networks to make the representations of two images, generated from the same original image but with different augmentations, close in terms of the L2 metric.
* The BYOL method achieves results that are comparable or even superior to supervised learning and other self-supervised methods.
* Applying a fixed pixel-wise permutation before running classification algorithms to distinguish between classes has proven to be particularly challenging.
* Inspired by BYOL, we have developed Permutated BYOL (PBYOL) as a solution for permuted image classification.
* The primary goal of the PBYOL algorithm is to provide a solution for the classification of permuted images. Additionally, the algorithm introduces a new approach that achieves better representation and can be applied to various vision tasks such as classification, detection, segmentation, and other tasks.
* The PBYOL model takes augmented permuted images as input, and its architecture is almost equivalent to the BYOL architecture.
  + PBYOL has succeeded in improving image representation and enhancing the accuracy of both permuted and non-permuted images.
  + The BYOL algorithm optimizes representation learning by ensuring that two images with different augmentations are considered equivalent in the embedding space, as they maintain the same semantic information. PBYOL extends this method by incorporating the task of understanding the permutation order, resulting in improved results compared to the BYOL method.
  + PBYOL achieves an improvement of ??% compared to standard training with ImageNet initialization on permuted images across different datasets, and it also outperforms the BYOL method.
  + Furthermore, PBYOL achieves a ??% improvement compared to standard training with ImageNet initialization on non-permuted images across different datasets.
  + The following table show the decrease in results as factor of grid size.
    - Important to notice that this data is simple, due to amount of class's, and therefore permutation effect on performance are soft, compare for the other datasets that will cover in experiments: CIFAR10, and Pets.

|  |  |  |
| --- | --- | --- |
| Dataset – dogs & cats | Accuracy | f-score |
| No perm | **0.994** | **0.994** |
| 2x2 | **0.983** | **0.981** |
| 3x3 | **0.978** | **0.978** |
| 4x4 | **0.956** | **0.956** |
| 5x5 | **0.946** | **0.946** |
| 10x10 | **0.846** | **0.85** |

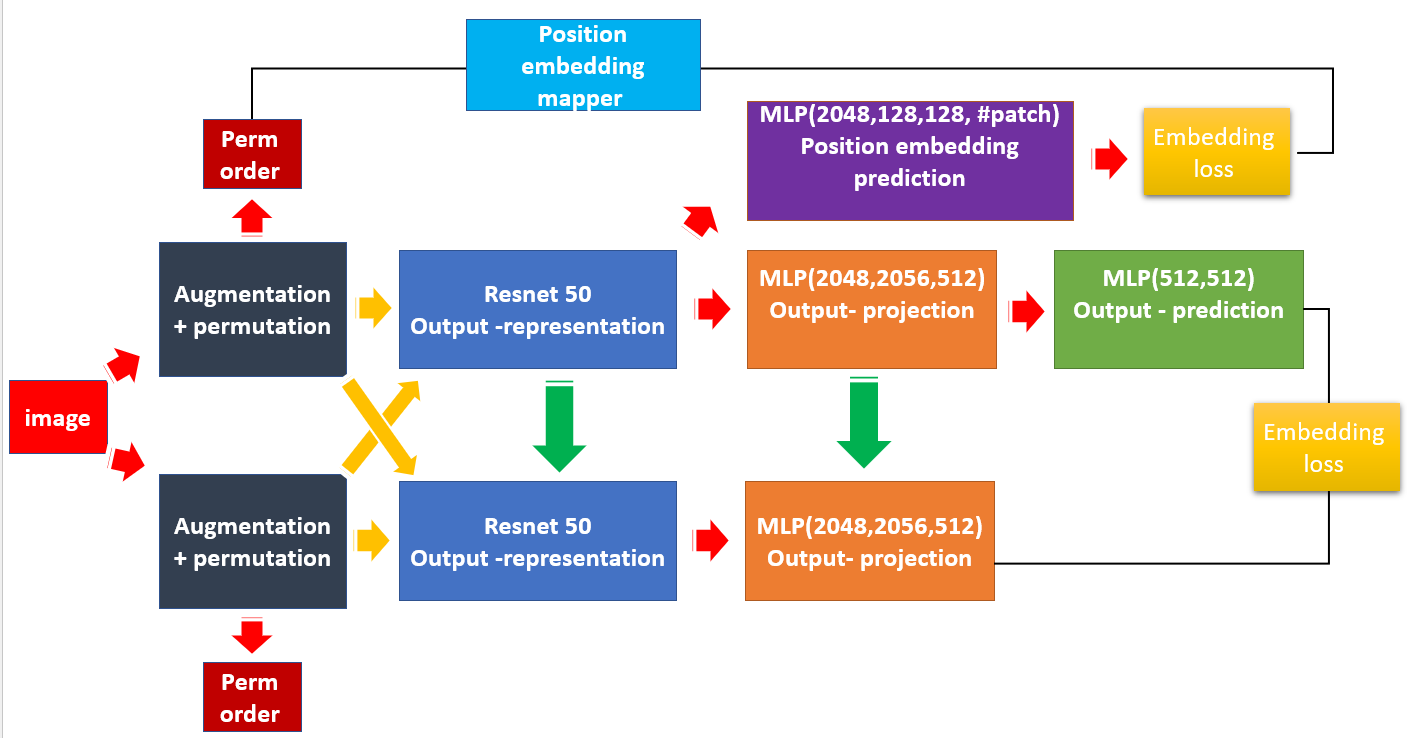
**Introduction:**

* In recent years, there has been an increased understanding that in order to achieve high performance, models need to improve their representation learning abilities, which in turn enhances model generalization and accuracy.
* Traditional approaches to learning good representations often rely on large, annotated datasets and specific architectures (such as CNNs or Transformers). However, obtaining such datasets with annotations can be expensive and sometimes ineffective. There is a growing need for users, ranging from university students to corporate workers, to achieve better representation without relying on large datasets, thereby improving model accuracy.
* Self-supervised learning (SSL) is one approach to tackle representation learning.
* SSL training typically involves two stages: the first stage involves training a model without labelled data, which results in improved model weight initialization. The second stage entails training the model with labelled data using the weights obtained in the first stage. Since the representation has already been improved compared to standard initialization methods like ImageNet, the model's performance is expected to be better than that of one-stage learning with ImageNet weight initialization. Moreover, the need for large, labelled datasets can be replaced with few-shot learning.
* Due to the advantages of SSL learning, this field has experienced significant growth in deep learning, with many researchers striving to make SSL techniques superior to supervised learning methods.
* Contrastive learning is an SSL technique that achieves high performance. However, it suffers from computational requirements. The technique enforces two positive samples (with the same semantic information) to be close in the embedding space (L2), while ensuring that negative samples are far from the positive samples. Another requirement of contrastive learning is to use a larger batch size, otherwise, the model may suffer from mode collapse, which mean poor performance.
* To address the disadvantages of contrastive learning, BYOL was developed. This algorithm eliminates the need for comparing positive and negative pairs and maintains performance even with a smaller batch size compared to contrastive learning methods.
* One of the goals of BYOL is to avoid the requirement of comparing positive to negative pairs. Instead, it focuses on forcing the representations of the same image, with two different augmentations, to be close in the representation space.
* In recent years, the success of the Transformers architecture has attracted the attention of many researchers. Transformers convert input data such as images, NLP, audio, etc., into tokens and process them using parallel computing. The main advantage of Transformers is their ability to understand the contextual relationships between tokens through attention mechanisms.
* Transformers handle tokens in parallel, but to maintain token order information, the model adds predefined position embeddings in NLP or optimizes position embedding parameters in vision.
* In this paper, we present PBYOL, a new method that can be considered a versatile approach for various tasks, with a focus on classification optimization. When a data scientist needs to generate a model for a specific classification task, the usual methodology involves using a strong encoder (CNNs or Transformers) with ImageNet weights and training on large, labelled datasets. The optimization process adapts the model weights from a well-trained ImageNet model to the specific classification task at hand.
* BYOL is composed of an online model and a target model, both of which share the same encoder (ResNet50) for generating representations and the same projection layer (MLP layers) for reducing the dimensions of the representations. The online model has an additional MLP layer for prediction placed between the online and target projections of the representation.
* PBYOL takes augmented permuted images as input. The permuted image is generated using a pre-defined permutation order, with each permutation having a unique index associated with all possible permutations. This index serves as the label for the positional embedding vector of the permutation.
* PBYOL aims to leverage this additional information for representation learning and uses the pre-defined permutation order as the positional embedding.
* PBYOL combines the optimization techniques of the BYOL model and the positional embedding of Transformers to create a powerful SSL technique. This additional optimization task enhances representation learning and improves model performance.
* In summary, our contributions can be summarized as follows:
  + Development of a new SSL technique.
  + Demonstrating the benefits of using augmented permutation images for SSL representation learning.
  + Integration of SSL and NLP position embedding into a single method.
  + Providing a solution to the drop in classification results when using permuted images.



BYOL

PBYOL



**Related work:**

* Unsupervised Learning:
  + "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzle" is a paper that focuses on solving jigsaw puzzles as a method of unsupervised learning.
  + According to this paper, a jigsaw puzzle is created by dividing an image into patches and randomly shuffling them.
  + The objective of this approach is to encourage the model to learn visual features that capture spatial relationships and contextual information present in the image. The model needs to understand the spatial arrangement of the patches, their visual coherence, and the relationships between adjacent patches.
  + Since this method is fully unsupervised, it is challenging to achieve the same level of performance or accuracy as supervised learning methods.
* Model Collapse:
  + One of the goals of SSL is to capture the full diversity and complexity of the underlying data distribution. However, model collapse in SSL refers to a situation where the model fails to learn good representations.
  + Model collapse occurs when the model focuses on a limited set of patterns or variations in the data, neglecting important aspects. This can result in a limited set of similar or repetitive outputs.
  + Model collapse in SSL learning can be caused by various reasons:
    - Insufficient Pretext Task Complexity: If the pretext task is not sufficiently complex or diverse, the model may converge to a limited set of patterns or variations in the data.
    - Biases in Pretext Task Design: The design of the pretext task can introduce biases that lead to mode collapse. If the pretext task favours certain types of patterns or features over others, the model may primarily focus on those patterns, ignoring other modes or variations. Biases can arise from the choice of augmentation techniques and the design of the network architecture.
  + To prevent mode collapse in SSL, the designer can take the following steps:
    - Diverse Pretext Tasks: Design pretext tasks that cover a wide range of variations and patterns in the data. Instead of relying on a single pretext task, consider using multiple tasks that encourage the model to capture different aspects of the data.
    - Hyperparameter Tuning: Adjust hyperparameters such as batch size or learning rate to discourage the model from focusing on specific patterns and encourage more generalization. However, increasing the batch size may lead to higher memory usage and longer training times.
    - Augmentation Techniques: Use diverse and sophisticated data augmentation techniques during training. By applying various augmentations, the model is encouraged to learn representations that are invariant to those augmentations, helping to capture diverse modes in the data.
    - Composite Loss Functions: Design loss functions that encourage diversity in the learned representations, such as multiple objectives or symmetric losses. These loss functions can encourage the model to capture diverse aspects of the data distribution.
* self-supervised:
  + SIMCLR (Similarity-based Contrastive Learning of Representations):
    - Contrastive learning is a technique that aims to learn meaningful representations from unlabelled data. It achieves this by encouraging similar instances to be close together in the learned feature space, while pushing dissimilar instances apart. The key idea behind contrastive learning is the use of positive and negative pairs to train a model to distinguish between different instances.
    - Positive pairs in contrastive learning consist of two augmented versions of the same input instance. These augmented versions are created by applying random transformations or perturbations to the original data. On the other hand, negative pairs are formed by taking two augmented versions of different instances.
    - contrastive loss:

Where:

– are feature vector

− is similarities measure

– is temperature factor

* + - SIMCLR has been successful in achieving high performance and improving representation learning. However, one limitation of SIMCLR is its high computational cost due to the large number of comparisons required during training. Additionally, in order to effectively train the model, it typically requires a high batch size to ensure convergence.
  + BYOL (Bootstrap Your Own Latent):
    - BYOL aims to address the need for comparing positive and negative pairs in contrastive learning.
    - The BYOL paper provides an explanation as to why it is possible to compare only positive examples and disregard the comparison to negative pairs.
    - As mentioned earlier, the BYOL architecture has been described. Now, we will delve into the training details, which are primarily relevant to the PBYOL method. It is important to note that during training, the online model is updated using Exponential Moving Average (EMA).
    - The EMA update is performed using the following equation:

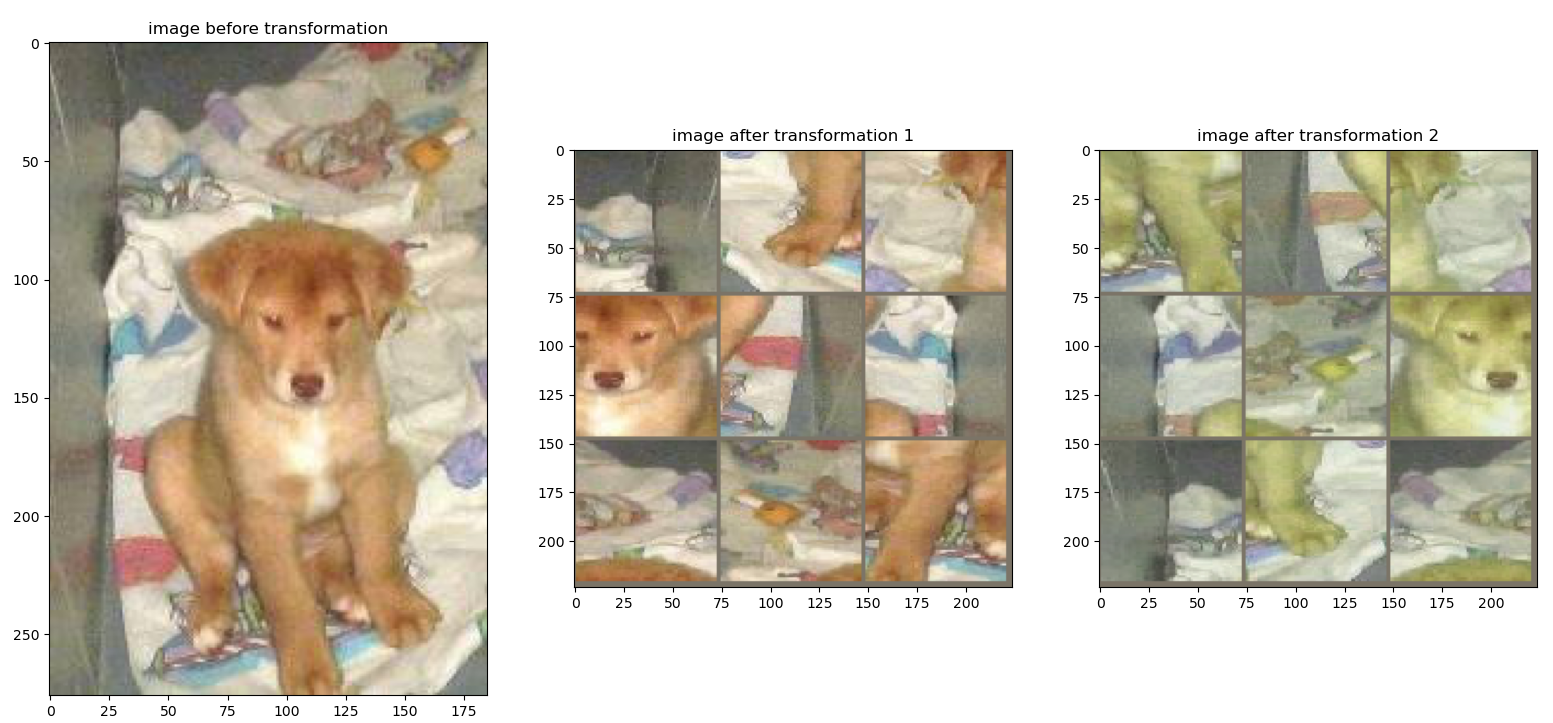
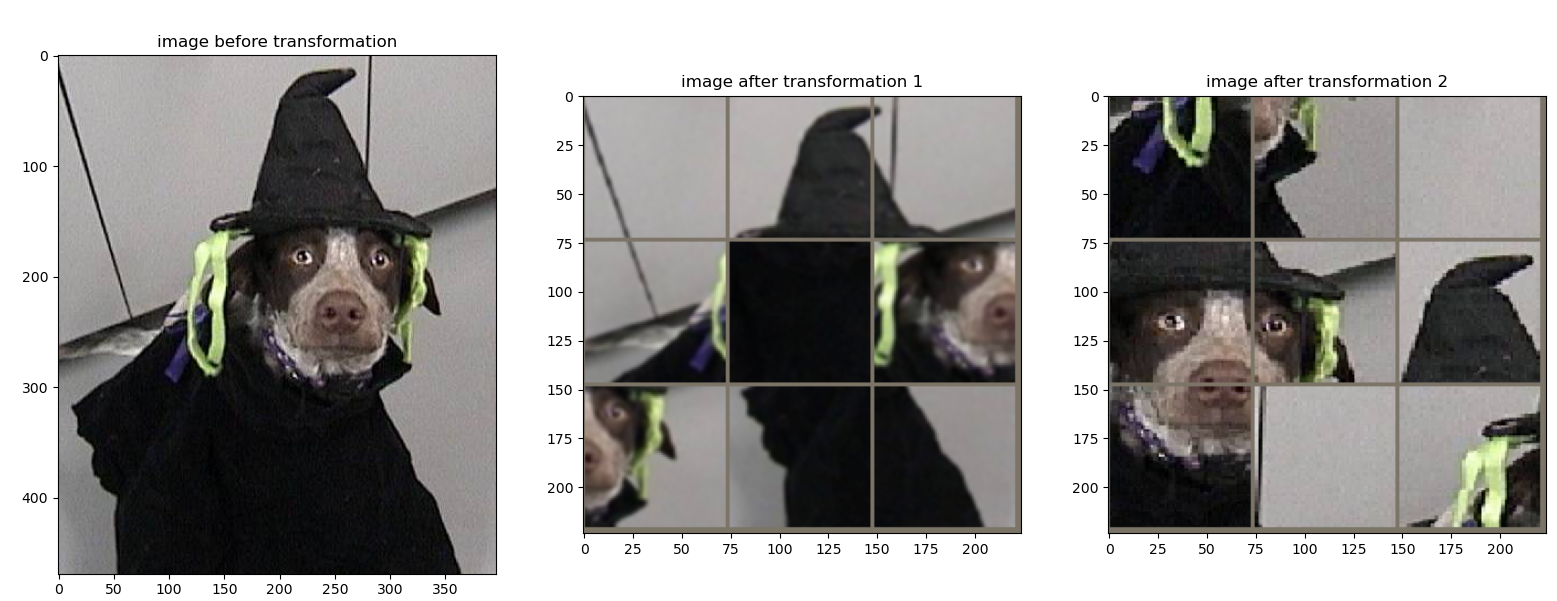
* + - – is number between 0 to 1 which define the memory of weight for , in BYOL paper experiments was prove that for BYOL succeed to prevent model collapse and learned good representation.
    - is not set for constant for all training, he is updating using the following formula:

k – is training step

K - is the overall amount of steps

– the initial

* + - In order that model will not fall into mode collapse, Online and target model are initialized randomly, with the goal that they will be enough different that model will not fall into mode collapse.
    - BYOL loss function is L2 between online model prediction output and target projection output, another method that the authors tried to prevent model collapse is by design symmetric loss:
      * l – is MSE between 2 vectors.
      * – are the outputs of online and target model.
* **Transformers**
  + The new era of transformers could be taught start from the paper attention is all you need, this paper present state of the art model for NLP tasks, present new units such as multi head attention and position embedding, those elements are some of the reasons for the transformer's success.
  + Position embedding required to stand with the following conditions:
* Order Preservation: Positional embedding is essential in preserving the order of tokens in the input sequence. Tokens that are closer in the input sequence should have position embeddings that reflect their proximity.
* Expressiveness: Position embeddings should possess sufficient expressiveness to encode a wide range of positional relationships, including both local and long-range dependencies. They should be able to capture the varying distances between tokens in the sequence.
* Dimensionality: Position embeddings should have a consistent and fixed dimensionality. The dimensionality should be carefully chosen to effectively represent the positional relationships between tokens in the input sequence.
* In the context of vision, positional embedding can be compared to patch-level tokenization in NLP. However, unlike NLP where the position embedding is typically pre-defined, in vision, the position embedding is treated as learnable parameters that the model needs to optimize.
* In both vision and NLP, the position embedding is added to the token embedding to provide additional information about the spatial or positional relations between different tokens or patches.



* Left image with horizontal flip.
* Right image with color jitter.
* Different permutation order
* Left image with gauss blur.
* Right image with random crop.
* Different permutation order

**Method:**

* In this section, we will describe the intuition behind the PBYOL method, its inputs, loss functions, and model units.
* Firstly, let's discuss our intuition. Similar to BYOL, the goal of PBYOL is to capture the fact that two images, undergoing different augmentations, contain the same semantic information and should be close in the embedding space. We adopt this approach.
* BYOL uses augmentations from a predefined set, with each augmentation being chosen by a predefined probability. These augmentations help the model generalize and capture the semantic information from two different images.
* Solving a puzzle by reconstructing an instance using separate puzzle parts, without knowing the puzzle's target, can be a challenging task for both young and adult individuals. The intuition behind solving such puzzles is to connect parts that, when combined, provide meaningful semantic information. For example, connecting two parts representing a person's eyes can indicate that the puzzle contains an image of a person.
* Capturing semantic information from separate puzzle parts is a complex task. If a model can understand the semantic relationships between unordered patches, it signifies that the model has a better representation. Image permutation can be considered as an image augmentation technique that further enhances the model's ability to generalize.
* Secondly, the permutation order is crucial, and having knowledge of the order can aid the model in achieving better representation. Like a detection task, where loss functions are composed of classification and regression losses, knowing that a specific region proposal does not contain an object helps the model focus on predicting more accurate bounding boxes for the actual objects to be detected. Therefore, the permutation order can assist in our optimization, and the PBYOL loss will consist of the BYOL loss when the images inserted into the online and target models are the permuted images. Additionally, the permutation order loss will be incorporated.
* All set of permutation can be thought as our dictionary, when we want that permutation which are close related ([1,2,3,4], [1,2,4,3]), will be closer in embedding space, and permutation that are not similar (([1,2,3,4], [4,1,3,2])), will be far from each other in embedding space.
* In order to achieve those attributes, we decided to use NLP position embedding from the paper attention is all you need, which take into consideration the max sentence size, the embedding space size, and the position in the sentence as our position embedding mapper which working base the following equation:
  + n – is the maximum size of sentence, in our case the number of possible permutations,
  + k – is the index of current permutation index in all possible permutation
  + d – amount of patch
  + i – index in the d vector size
* BYOL model inputs are:
  + – image sample
  + – related to online and target model in this order.
  + – x image pass into augmentation from set of augmentation.
  + – augmented images pass into permutation augmentation.
  + – permutation order of .
  + - position embedding for permutation order. Those are the labels for position embedding branch.
  + – representation of , the output of encoder (feature extraction)
  + – projections of representation vector, the output of MLP layer.
  + - prediction ofprojected representations, in online model, the output of MLP layer.
  + - prediction of position embedding using , the output of MLP layer.
* Loss function


  + – balance factor, if equal to 0 is same as BYOL
    - tend to decrease to 0 🡪 balanced factor required to balanced between the 2 loss.
* The loss function enjoys from BYOL loss attribute and adjust the symmetric loss for the position embedding loss.
* Model description:
  + Encoder - ResNet 50, composed from 4 blocks, the outsize is 2048.
    - Initialized with image net wights in online and target model.
  + Projection unit – MLP unit, convert from size 2048🡪2056🡪512.
    - Initialized with different random in online and target model.
  + Prediction unit - MLP unit, convert from size 512🡪512.
    - Initialized with different random in online model.
  + Prediction PE - MLP unit, convert from size 2048🡪1024🡪512🡪amount of patch.
    - Initialized with different random in online model.
  + In BYOL the prediction layer was set to size 256, and author mentioned that less than that the vector is not expressive enough, because our augmentation is more complex, we increase the size into 512.
  + During our experiment, we conclude that best place to predict the position embedding is from the last layer with needed to symbolize the sematic information.
  + we freeze all encoder weights, except layer 4.2 which is the last encoder layer 🡪 overall 12M trainable parameters from 32M parameters.
  + The target model is not trainable, and model weights are updated using EMA from online model to the target model.

**תמונה שמכילה טקסט, צילום מסך, גופן, מלבן

התיאור נוצר באופן אוטומטי**

**Pre-processing:**

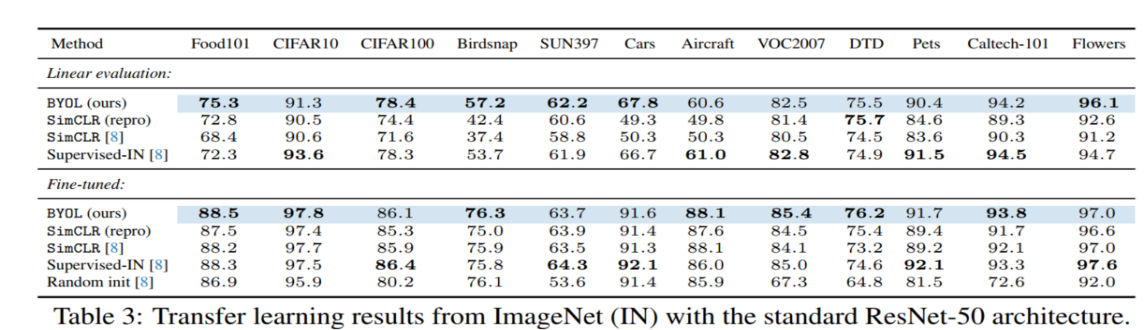
* Data sets:
  + Dog and cats:
    - Number of samples 25K
    - Contain 2 class.
  + Pets
    - Number of samples 7376
    - Contain 37 class.
  + CIFAR10
    - Number of samples 50K
    - Contain 10 class.
* Image augmentation set:
  + Resize image to 224x224.
  + One from the following augmentation set:
    - rand crop.
    - colour jitter.
    - random horizontally flip.
    - gaussian blur.

**Training details**

* Optimizer:
  + Stage 1 – representation learning, learning rate 1e-4.
  + stage 2 – supervised learning, learning rate 1e-3.
* Regularization
  + Early stopping allowed 9 epochs without change.
* Batch size 512 – max possible due computational abilities.
* Embedding space dimensions – 512.
* balanced factor = 1
* Loss functions for position embedding space and for representation projected space is L2.
* The metric for evaluating our performance accuracy and F1-score.
* In order to be concise, for the validation dataset, the permutation order was sets once at the start of training, compare to training data.

**Experimental evaluation:**

* BYOL results on classification task.



* Many experiments needed to be done 😊
* Results until this document:
  + CIFAR10 using 10% of data, using grid 3x3:
    - Results on train data permuted images:

|  |  |  |
| --- | --- | --- |
| Dataset – CIFAR10 | Accuracy | f-score |
| FULL SUPERVISED | **0.764** | **0.729** |
| PBYOL | **0.827** | **0.827** |

* + - Results on train data on original images:

|  |  |  |
| --- | --- | --- |
| Dataset – CIFAR10 | Accuracy | f-score |
| FULL SUPERVISED | **0.795** | **0.796** |
| PBYOL | **0.789** | **0.802** |

* + Pets using all data, using grid 3x3:
    - Results on validation data permuted images:

|  |  |  |
| --- | --- | --- |
| Dataset – CIFAR10 | Accuracy | f-score |
| FULL SUPERVISED | **0.55** | **0.74** |
| PBYOL | **0.827** | **0.827** |

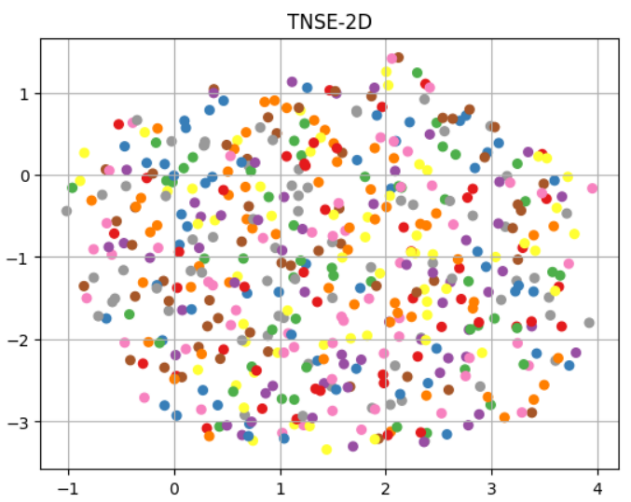
* + Pets using all of data, using grid 4x4:
    - Results on train data permuted images:

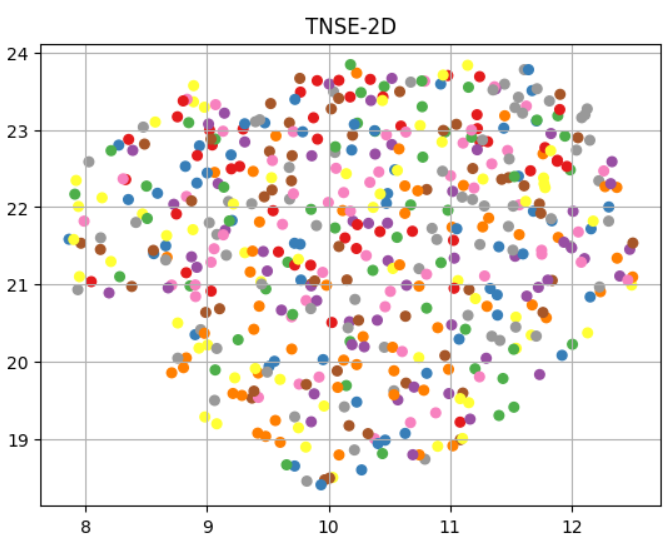
|  |  |  |
| --- | --- | --- |
| Dataset – CIFAR10 | Accuracy | f-score |
| FULL SUPERVISED | **0.444** | **0.309** |
| PBYOL | **0.52** | **0.444** |

* + - Results on train data on original images for 15 epochs:

|  |  |  |
| --- | --- | --- |
| Dataset – CIFAR10 | Accuracy | f-score |
| FULL SUPERVISED | **0.767** | **0.705** |
| PBYOL | **0.814** | **0.777** |

* Effect of training after dimension reduction:
  + ImageNet weights:



* + after PBYOL stage 1:
  + Can be notice about the variance after PBYOL is higher
* Experiment could be done:
  + Needed to decide on which datasets we wish to do experiments.
  + Results on partial amount of data: 0.1,0.25,0.5,0.75,0.9,1
  + Balance factor: 0 (BYOL), 1,10,100
  + Learning rate and batch size
  + Model weight initialization
  + EMA updates
  + Best permutation grid size effect on results also on permutation dataset and on original dataset
  + Maybe to choose other loss than L2, like L1\L1SMOOOTH
  + Effect on representation after stage1 using dimension reduction.
  + Reconstruct image using position embedding prediction.

**Conclusion:**

* PBYOL is good 😊