DSE313 Assignment: Phase 1 Report

Feature Selection for few-shot building instance classification using Genetic Algorithms

Group 10

Gandhi Ananya Amit (20319)

Devashish Tripathi (21093)

Kartikey Singh (20146)

Saksham (20240)

Abstract

In the urban landscape, buildings with both a usage specified to them(i.e. single-type) and those that serve multiple purposes(i.e. mixed-type) are common. While it is easy to map single-type buildings, it proves difficult to do so for mixed-types. This work proposes the use of Genetic Algorithms and Contrastive Learning to classify building usage types from Street-View images. For this phase: i) a dataset of mixed-type buildings is created using Google Street-View and Equirectangular Toolbox, and ii) Two models, ResNet-50 and DINOv2, are evaluated on a given dataset of single-type buildings. Among the two models, DINOv2 achieved far superior results.

1. Introduction

In developing countries like India, it is common to see one single building housing various different ventures on each floor. For example, buildings with shops on the ground floor and residential arrangements on other floors are common. So are buildings under construction with inhabitants on completed floors and buildings, with parts of them being used for commercial purposes while the rest of them are abandoned. Marking these 'mixed-type' buildings is much more difficult as compared to buildings dedicated to a specific purpose, the so-called 'single-type' buildings.

A major work done in this field is [1], which used *Street-View* images to classify the use type of single-type buildings and marked the building usage on an aerial view image of the locality from which the *Street-View* images were taken. For this task, *Street-View* images were taken, outliers were removed using X, and four models, ResNet18, ResNet34, AlexNet and VGG16, were experimented and fine-tuned with for the classification task. VGG16 was finally selected based on the best F1 scores and accuracy.

This project involves using the combination of Genetic Algorithms and Contrastive Learning in addition to pre-existing models to create a method to classify mixed-type buildings using *Street-View* images.

Genetic Algorithms work similarly to natural selection, where parts of two genes are merged, a mutation may occur, and out of a population, the strongest genes are allowed to mate, leading to a new population. In the problem of Floor-Usage type classification, Genetic Algorithms may be used to ascertain whether a floor is of a particular type. Several predictions of the floor usage may be possible, which would act as the population from which the algorithm would work by selecting the strongest likely floor type. It may also be used to segment individual floors based on boundaries and the strongest likely floor number.

Contrastive Learning is a Self-Supervised Learning paradigm used to combat the requirement for a high amount of labelled data for training CNNs. This method works similarly to kNN clustering in the sense that the data points that match are clustered closer while the negative matches are distanced. It allows the model to cluster the data points based on the high-level features and, thus, reduces the need for labelled data. Using this would allow the *Street-View* images directly, and Genetic Algorithms could be used in combination to allow faster and more precise clustering of the images.

The first phase of this project involves creating a dataset of Street-View images of mixed-type buildings, followed by a review of two CNN models, ResNet-50 and DINOv2, on a dataset of single-type buildings divided across seven classes: Abandoned, Commercial, Industrial, Religious, Residential, Under-Construction and Others. The results of both models are compared and discussed in this report.

2. Methodology

The following methodology was utilized in the workflow of this phase:

- **Image Collection:** *Google Street View* was used for the purpose of locating the ideal mixed-type buildings for the mixed-type buildings dataset. Once located, the images were downloaded using the software *Street View Download 360*. The downloaded images, which were in equirectangular format, were converted to a normal FOV form using the *equirectangular-toolkit*, which can be found at [2]. Sample mixed-type building images are in Figure 1. The dataset can be found at this link: Here

- **Model evaluation:** Once the dataset was created, two pre-trained models, ResNet-50[4] and DINOv2 [3], which were trained on the ImageNet dataset were selected. For this phase of the project, they were evaluated on a dataset of single-type buildings across seven classes: Abandoned, Commercial, Industrial, Religious, Residential, Under-Construction and Others. The Dataset consists of 2595 Training images and 391 Testing images in total.

  Transfer Learning was used to train the models. The models were frozen as only the final layer was to be trained. A final layer, which had an output of 7 classes, was added, which was to be trained. 25% of the Training Data was selected for Validation.

Figure 1. *Sample mixed-type images*

The models were evaluated with the help of metrics such as Confusion Matrix, Training and Validation Accuracy and Loss Curves. Adam optimizer was used for both the models, with a learning rate of 5×10^{-4}. Cross Entropy Loss was used for both models. ResNet-50 was trained for 25 epochs, while DINOv2 was trained for 15. DINOv2 was trained for fewer epochs because an initial training done on 25 epochs resulted in the Validation loss increasing after 12 epochs, thus implying the presence of overfitting. The training statistics for this initial run can be found in the appendix.

3. Results

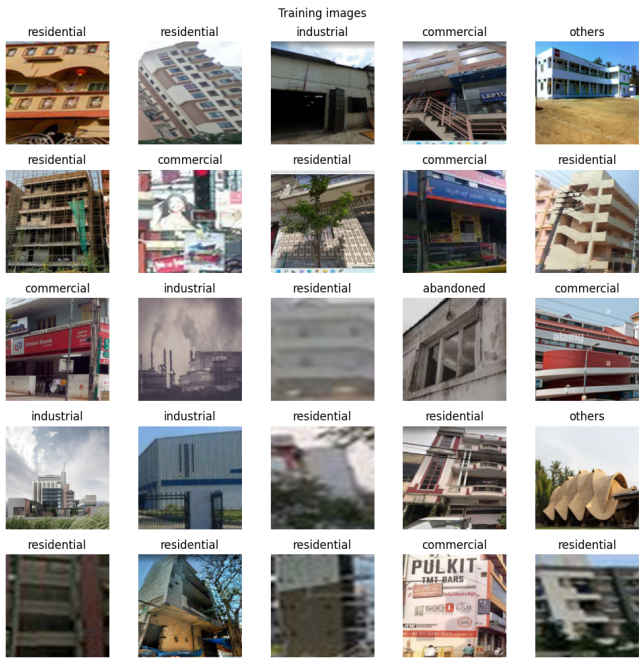
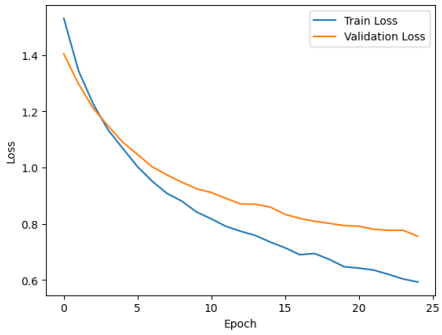


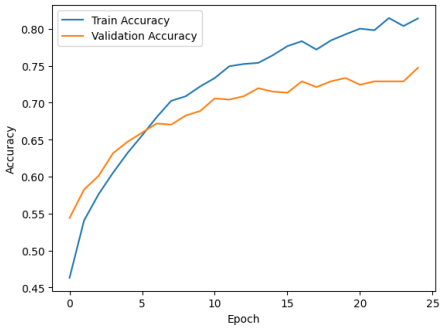
Figure 2. *A sample of images used for training along with their actual labels*

3.1. ResNet-50:

The relevant plots and figures can be found in Figure 3 and Figure 4.



(a) *Loss curves*



(b) *Accuracy curves*

Figure 3. *Relevant plots for ResNet-50(1/3)*

3.2. DINOv2:

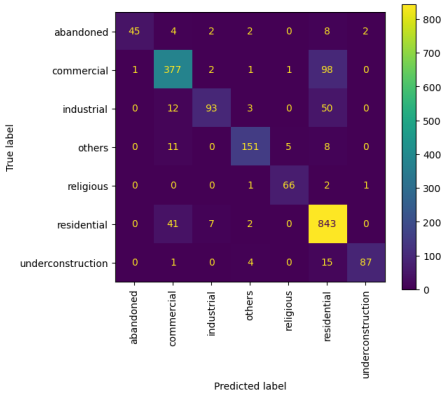
The relevant plots and figures can be found in Figure 5 and Figure 6.

4. Discussions

Based on the results, the performance of DINOv2 is much more superior as compared to ResNEt-50. Before discussing a possible explanation of why this may be the case, we first present a reason for placing importance on a specific class.

For this specific problem, as it can be noticed, the dataset contains a very high number of residential building images. Even while creating our dataset for the mixed-type buildings, most of the buildings in the Indian context were commercial on the lower floors and residential on the upper. This is significant as a majority of mixed-type Indian buildings, due to various factors such as population density, are going to have a residential or a commercial component. Industrial, Religious and Abandoned buildings would, thus, rarely have a presence in a mixed-type dataset. Also, it is quite common, as can be seen in some images of the dataset, that a few Abandoned buildings have commercial segments. As such, we consider that the class of Commercial buildings should be considered important as for the Indian context, they are the most likely to be found in mixed-type combinations.

Before explaining our reasoning for why DINOv2 performs better than ResNet-50, we would like to discuss both models in brief.

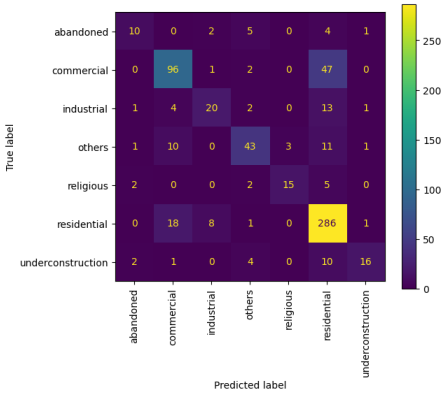


(c) *Confusion Matrix for Training*

Classification Report:

	precision	recall	f1-score	support
abandoned	0.98	0.71	0.83	63
commercial	0.85	0.79	0.81	480
industrial	0.89	0.59	0.71	158
others	0.92	0.86	0.89	175
religious	0.92	0.94	0.93	70
residential	0.82	0.94	0.88	893
underconstruction	0.97	0.81	0.88	107
accuracy			0.85	1946
macro avg	0.91	0.81	0.85	1946
weighted avg	0.86	0.85	0.85	1946

(d) *Classification Report for Training*



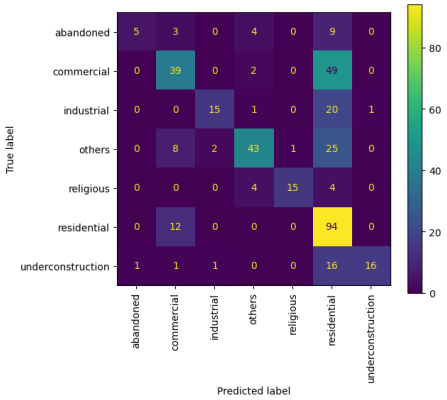
(e) *Confusion Matrix for Validation*

Classification Report:

	precision	recall	f1-score	support
abandoned	0.62	0.45	0.53	22
commercial	0.74	0.66	0.70	146
industrial	0.65	0.49	0.56	41
others	0.73	0.62	0.67	69
religious	0.83	0.62	0.71	24
residential	0.76	0.91	0.83	314
underconstruction	0.80	0.48	0.60	33
accuracy			0.75	649
macro avg	0.73	0.61	0.66	649
weighted avg	0.75	0.75	0.74	649

(f) *Classification Report for Validation*

Figure 3. *Relevant plots for ResNet-50(2/3)*



(g) *Confusion Matrix for Testing*

Classification Report:

	precision	recall	f1-score	support
abandoned	0.83	0.24	0.37	21
commercial	0.62	0.43	0.51	90
industrial	0.83	0.41	0.55	37
others	0.80	0.54	0.65	79
religious	0.94	0.65	0.77	23
residential	0.43	0.89	0.58	106
underconstruction	0.94	0.46	0.62	35
accuracy			0.58	391
macro avg	0.77	0.52	0.58	391
weighted avg	0.68	0.58	0.58	391

(h) *Classification Report for Testing*

Figure 3. *Relevant plots for ResNet-50(3/3)*

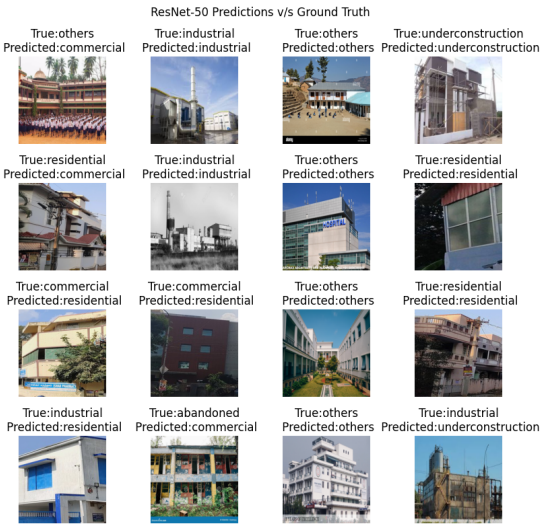


Figure 4. *Predictions vs Ground Truth for ResNet-50*

4.1. ResNet50 Architecture

ResNet50 Framework

The ResNet50 model is a Supervised Learning Model which has been trained the ImageNet-1K dataset, which in itself is a widely used dataset for training and benchmarking image classification models. This dataset contains over a million images labelled with 1000 different classes.

It is a variant of the Residual Network architecture, a popular convolutional neural network (CNN) architecture and is named so because it contains 50 layers deep.

This architecture typically consists of several stages of convolutional layers, each followed by batch normalization and ReLU activation.

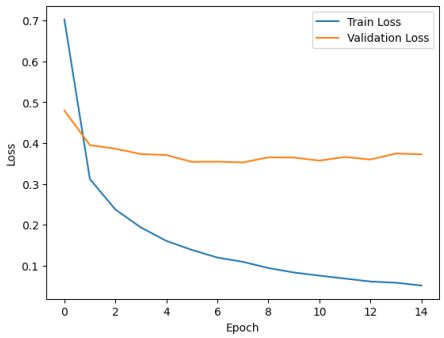
The key innovation of ResNet50 is that it comprises of 'skip-connections' or 'residual-connections' which add the input of a block to its output before applying the activation function, helping to mitigate the Vanishing Gradient Problem. This enables ease and guarantees of a higher accuracy in training of much deeper networks.

$$\text{output} = \text{activation}(\text{input} + F(\text{input}))$$

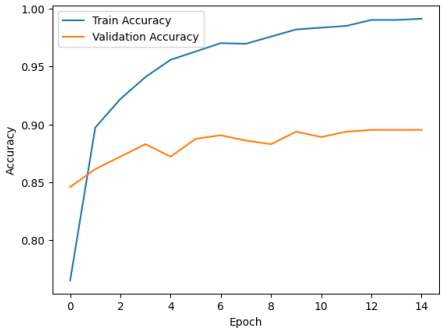
$F(\text{input})$: transformation learned by the block of layers input: input to the block output: output of the block

Architecture Summary

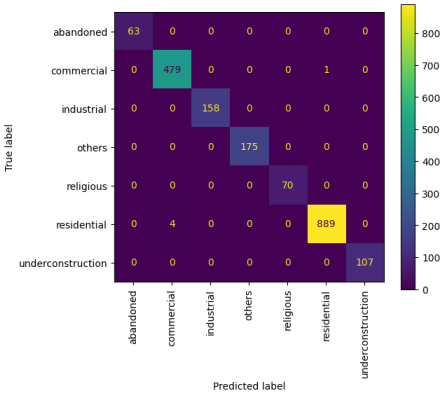
The core architecture of this ResNet50 model involves:



(a) *Loss curves*



(b) *Accuracy curves*



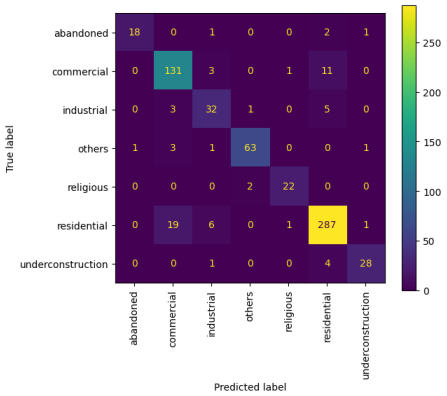
(c) *Confusion Matrix for Training*

Classification Report:

	precision	recall	f1-score	support
abandoned	1.00	1.00	1.00	63
commercial	0.99	1.00	0.99	480
industrial	1.00	1.00	1.00	158
others	1.00	1.00	1.00	175
religious	1.00	1.00	1.00	70
residential	1.00	1.00	1.00	893
underconstruction	1.00	1.00	1.00	107
accuracy			1.00	1946
macro avg	1.00	1.00	1.00	1946
weighted avg	1.00	1.00	1.00	1946

(d) *Classification Report for Training*

Figure 5. *Relevant plots for DINOv2(1/2)*

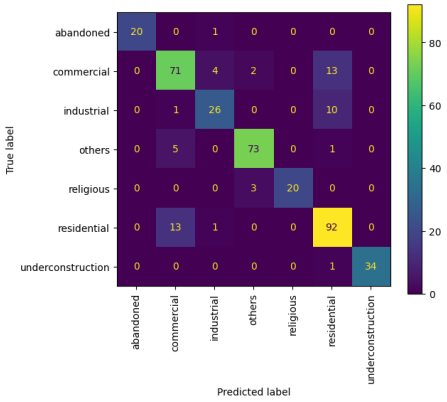


(e) *Confusion Matrix for Validation*

Classification Report:

	precision	recall	f1-score	support
abandoned	0.95	0.82	0.88	22
commercial	0.84	0.90	0.87	146
industrial	0.73	0.78	0.75	41
others	0.95	0.91	0.93	69
religious	0.92	0.92	0.92	24
residential	0.93	0.91	0.92	314
underconstruction	0.90	0.85	0.88	33
accuracy			0.90	649
macro avg	0.89	0.87	0.88	649
weighted avg	0.90	0.90	0.90	649

(f) *Classification Report for Validation*



(g) *Confusion Matrix for Testing*

Classification Report:

	precision	recall	f1-score	support
abandoned	1.00	0.95	0.98	21
commercial	0.79	0.79	0.79	90
industrial	0.81	0.70	0.75	37
others	0.94	0.92	0.93	79
religious	1.00	0.87	0.93	23
residential	0.79	0.87	0.83	106
underconstruction	1.00	0.97	0.99	35
accuracy			0.86	391
macro avg	0.90	0.87	0.88	391
weighted avg	0.86	0.86	0.86	391

(h) *Classification Report for Testing*

Figure 5. *Relevant plots for DINOv2(2/2)*

- Convolutional layers (these are structured to use residual connections)
- Batch normalization and ReLU activations after convolutional layers
- A global average pooling layer (GAP) followed by a fully connected layer (FC)

4.2. DINOv2: DINO Framework

The DINOv2 Training Model is an advanced self-supervised learning model for Vision Transformers (ViTs).

DINO stands for "DIstillation with NO labels". It doesn't require labelled data to learn robust visual features.

The framework of DINO allows training vision transformers by using knowledge distillation, where a teacher network's outputs are used to understand and classify visual content without relying on annotated datasets. Instead, these models can learn to recognize patterns and features directly from the structure of the data.

The DINO model that we used has been trained on the

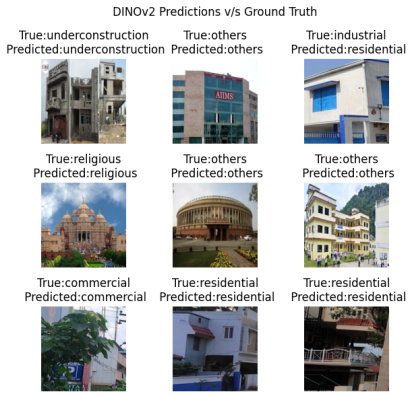


Figure 6. *Predictions vs Ground Truth for ResNet-50*

ImageNet-1k dataset, which comprises roughly 1 million images across 1,000 classes. The DINOv2 method represents a shift towards models learning from data without the need for explicit annotations.

This approach is particularly significant because it reduces reliance on large labelled datasets, which can be costly and time-consuming to produce. Moreover, learning features without supervision can lead to models that generalize better to a wider range of tasks since they're not limited to the specific biases and constraints of labelled datasets.

Why DINO is outperforming ResNet50 in our case? DINO, being a self-supervised learning model, results in more accuracy on testing and training more "flexible" data that is data which is considered not to have any models majorly trained upon. DINO uses the attention mechanism to zoom in on various areas of the image, which results in more detailed feature representations. Because DINO learns from varied augmentations and is not confined to specific labelled data, it also tends to learn more robust and generalizable features. This leads to better performance on diverse datasets and in different visual domains.

On the other hand, ResNet50 architectures relying on supervised learning require labeled datasets to learn. This limits the model to the distribution of the training data and hence, it does not generalize well outside of that specific dataset. Our data heavily consists of Indian Building and their types predominantly seem to be lying outside of this specific training dataset.

ResNet being a type of CNN, its convolutional nature also tends to focus on local patterns rather than the global context as opposed to our DINO Training model.

5. Conclusion

For our provided Dataset on the single-type buildings across seven classes: Abandoned, Commercial, Industrial, Religious, Residential, Under- Construction and Others. The Dataset consists of 2595 Training images and 391 Testing images in total, we run two pre-trained models: i) ResNet-50 and ii) DINOv2.

With 25% of the Training Dataset being splitted to be Validation Data Set and Testing both the models on the given Testing Dataset, we get the following accuracy results for both the models:

i) ResNet-50: 58%

ii) DINOv2: 86%

Hence, we conclude that DINO's better performance over RESNET50 might be due to the attention mechanism in ViTs that allows the model to weigh different parts of the training data more precariously as opposed to the focus on local patterns by ResNet50.

Another reason DINO gives better performance may be due to the versatility in our data of Indian Buildings which strike close to real-world applications over which DINO produces more robust and globally contextual features due to it being a self-supervised training model as opposed to ResNet50 which has learnt features that are more specialized to it's pre-trained dataset and tasks, which limits its performance on data that differs significantly from the original training one.

With a strikingly higher precision for the DINOv2 model as compared to the ResNet-50 Training Model for single-type Indian buildings, we conclude we use in our further phase of determining and classifying of mixed-type Indian buildings, the DINOv2 model for training the collected datset of the 100 mixed-type Indian building images.

References

[1] Jian Kang et al. "Building instance classification using street view images". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (2018). Deep Learning RS Data, pp. 44–59. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2018.02.006. URL: https://www.sciencedirect.com/science/article/pii/S0924271618300352.

[2] Nitish Mutha. *GitHub - NitishMutha/equirectangular-toolbox: Handy tool for equirectangular images — github.com*. https://github.com/NitishMutha/equirectangular-toolbox.

[3] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2023. arXiv: 2304.07193 [cs.CV].

[4] Ross Wightman, Hugo Touvron, and Herve Jegou. "ResNet strikes back: An improved training procedure in timm". In: *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*.

A. Original DINOV2 training:

This version of the model started showing signs of overfitting. Thus, the number of epochs was reduced from 25 to 15.

```
Epoch:1  Train_Loss: 0.6491    Val_Loss: 0.4596    Train_Acc: 0.7831    Val_Acc: 0.8567    Time: 40.76 seconds
Epoch:2  Train_Loss: 0.3153    Val_Loss: 0.4054    Train_Acc: 0.8880    Val_Acc: 0.8644    Time: 37.49 seconds
Epoch:3  Train_Loss: 0.2380    Val_Loss: 0.3816    Train_Acc: 0.9188    Val_Acc: 0.8814    Time: 39.46 seconds
Epoch:4  Train_Loss: 0.1927    Val_Loss: 0.3749    Train_Acc: 0.9368    Val_Acc: 0.8829    Time: 43.49 seconds
Epoch:5  Train_Loss: 0.1613    Val_Loss: 0.3646    Train_Acc: 0.9491    Val_Acc: 0.8814    Time: 36.73 seconds
Epoch:6  Train_Loss: 0.1358    Val_Loss: 0.3547    Train_Acc: 0.9630    Val_Acc: 0.8875    Time: 36.86 seconds
Epoch:7  Train_Loss: 0.1198    Val_Loss: 0.3499    Train_Acc: 0.9687    Val_Acc: 0.8983    Time: 38.80 seconds
Epoch:8  Train_Loss: 0.1039    Val_Loss: 0.3540    Train_Acc: 0.9733    Val_Acc: 0.8952    Time: 38.25 seconds
Epoch:9  Train_Loss: 0.0920    Val_Loss: 0.3592    Train_Acc: 0.9774    Val_Acc: 0.8952    Time: 38.27 seconds
Epoch:10 Train_Loss: 0.0821    Val_Loss: 0.3567    Train_Acc: 0.9815    Val_Acc: 0.9045    Time: 37.39 seconds
Epoch:11 Train_Loss: 0.0743    Val_Loss: 0.3584    Train_Acc: 0.9825    Val_Acc: 0.9060    Time: 36.63 seconds
Epoch:12 Train_Loss: 0.0682    Val_Loss: 0.3674    Train_Acc: 0.9846    Val_Acc: 0.8998    Time: 39.90 seconds
Epoch:13 Train_Loss: 0.0615    Val_Loss: 0.3622    Train_Acc: 0.9897    Val_Acc: 0.9014    Time: 38.75 seconds
Epoch:14 Train_Loss: 0.0563    Val_Loss: 0.3733    Train_Acc: 0.9908    Val_Acc: 0.8968    Time: 38.80 seconds
Epoch:15 Train_Loss: 0.0506    Val_Loss: 0.3719    Train_Acc: 0.9938    Val_Acc: 0.9029    Time: 42.59 seconds
Epoch:16 Train_Loss: 0.0481    Val_Loss: 0.3825    Train_Acc: 0.9923    Val_Acc: 0.8968    Time: 40.27 seconds
Epoch:17 Train_Loss: 0.0443    Val_Loss: 0.3873    Train_Acc: 0.9964    Val_Acc: 0.8968    Time: 41.41 seconds
Epoch:18 Train_Loss: 0.0417    Val_Loss: 0.3795    Train_Acc: 0.9959    Val_Acc: 0.9014    Time: 42.61 seconds
Epoch:19 Train_Loss: 0.0371    Val_Loss: 0.3930    Train_Acc: 0.9974    Val_Acc: 0.8968    Time: 34.97 seconds
Epoch:20 Train_Loss: 0.0357    Val_Loss: 0.3835    Train_Acc: 0.9974    Val_Acc: 0.9076    Time: 39.81 seconds
Epoch:21 Train_Loss: 0.0330    Val_Loss: 0.3948    Train_Acc: 0.9979    Val_Acc: 0.8998    Time: 42.87 seconds
Epoch:22 Train_Loss: 0.0301    Val_Loss: 0.3936    Train_Acc: 0.9979    Val_Acc: 0.8998    Time: 38.81 seconds
```

Figure 7. *Metrics in the original training of DINOv2*

B. Contributions:

For Phase 1, the contributions per task are as follows:

- **Dataset Collection**: All team members collected the required 25 images.
- **ResNet-50 training**: Devashish and Ananya
- **DINOv2 training**: Devashish, Saksham and Kartikey
- **Report Writing**: Ananya and Devashish