VISION TRANSFORMERS FOR SONAR IMAGE CLASSIFICATION

N Warakagoda Norwegian Defence Research Establishment (FFI), Kjeller, Norway
Ø Midtgaard Norwegian Defence Research Establishment (FFI), Kjeller, Norway

1 INTRODUCTION

We consider the task of object classification in synthetic aperture sonar (SAS) images. Classification of objects in SAS images based on shape and material properties is a vital task in underwater automatic target recognition (ATR) and hence in applications such as mine counter measure systems (MCM).

Deep learning has been a highly successful technique for classification of optical images[4,8]. Subsequently, the technique has been adopted in many domains including object classification in SAS images[12,11,5]. Initial success of deep learning based image classification was achieved with Convolutional Neural Networks (CNNs)[4,8], and it has been the preferred technique in the field for several years. However, in the recent years a neural network architecture known as Transformers emerged as a better performing approach to image classification. Transformers were proposed initially for sequence processing tasks such as machine translation[10], but later adapted to image classification in the form of so called Vision Transformers (ViT)[2].

ViTs are generally superior to CNNs when the training dataset is large enough and sufficient computational resources (memory and FLOPs) can be allocated. The superiority can be observed in terms of classification accuracy, as well as robustness to noisy data and adversarial attacks[6]. However, ViTs have less inductive bias and hence more data and/or stronger regularization may be a precondition for good performance.

In this work we apply ViTs to the classification task mentioned above. The main goal of the work is to perform an initial comparative investigation of the ViTs against the CNNs in the context of SAS images, especially with a relatively small amount of training data.

The organization of the paper is as follows: Section 2 describes the background techniques relevant for this work. Section 3 gives a description of the task including the dataset and training/evaluation procedures. Details of the experiments and results are presented in Section 4. Finally in Section 5 conclusions are drawn.

2 BACKGROUND

2.1 Convolutional Neural Networks

Deep Learning has been an immensely successful approach in many tasks related to artificial intelligence including image classification. CNNs has been the prominent model that has contributed to this success. There are a number of well known CNN architectures that are pre-trained with optical images and usually available in the public domain. One such architecture is *Inception-Resnet-v2*[8] and it is used in our object classification experiments. The reason for this choice is that it was among the best in optical image classification while keeping the number of parameters and computational cost at a reasonable level.

We also apply *transfer learning*, meaning that the pre-trained neural network on optical images of the ImageNet data set[7] is trained again on our own SAS image data.

2.2 Vision Transformers

As is the case of regular transformers, ViTs are based on the *self-attention* mechanism[10]. Consider an input sequence of N number of D-dimensional vectors arranged in a matrix $\mathbf{z} \in \mathbb{R}^{N\times D}$. In order to perform self attention operation on this sequence, we need to split/transform the sequence into three quantities $\mathbf{q}, \mathbf{k}, \mathbf{v}$ called the triplet of *query, key* and *value*. This is done by multiplying the input with a trainable matrix $\mathbf{U}$ as follows:

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}, \text{ where } \mathbf{U} \in \mathbb{R}^{D\times 3D_h}.$$

As can be deduced from the above equation, each quantity $\mathbf{q}, \mathbf{k}$ and $\mathbf{v}$ is a sequence of N vectors of dimensions D_h arranged as a matrix of dimensions $N \times D_h$. Next step of the self-attention operation is calculation of the attention weights

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{D_h}}\right).$$

It is clear that $A \in \mathbb{R}^{N\times N}$. Finally, the self-attention (SA) is calculated by multiplying the *value* sequence by the attention weights:

$$\text{SA}(\mathbf{z}) = \mathbf{A}\mathbf{v}.$$

Usually, several self-attention operations are performed in parallel, results are concatenated and projected back to a D-dimensional space. This process is known as *multi-head attention* (MSA) and can be defined by the following formula:

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(\mathbf{z}); \text{SA}_2(\mathbf{z}); \cdots; \text{SA}_k(\mathbf{z})]\mathbf{U}_m, \text{ where } \mathbf{U}_m \in \mathbb{R}^{kD_h\times D}$$

With the multi-head attention operation on a sequence of vectors defined as above, Transformer Encoder can be constructed as shown on the right hand side of Figure 1. In this case we also need to employ other more conventional neural network operations, layer nomalization (Norm) and multi-layer perceptron (MLP).

Vision Transformer is just a way of using the Transformer Encoder on a serialized image as shown in the left hand side of Figure 1. The main idea here is to divide a given input image into a set of patches and create a sequence of using these patches. Each image patch is flattened to form a vector, so that it fits into

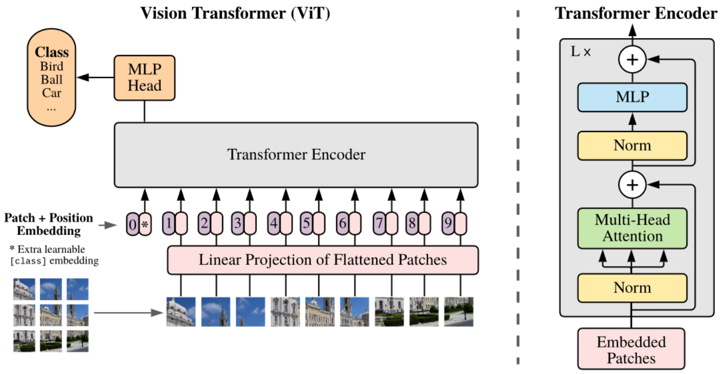


Figure 1: Vision Transformer architecture

standard definition of self-attention operations. Proponents of ViT also chose to project the image patch vectors to a D-dimensional space and add positional information to them to form the final input vector sequence. The Transformer Encoder is fed with this input sequence and an output vector sequence is generated. We need only the first vector of the output sequence to calculate the class probabilities using an MLP head.

In our experiments we used one of the first ever ViT models known as *ViT-Base-patch16-224*[2] as well as three variants of a model known as *DeiT*[9]; *DeiT-Base-patch16-224*, *DeiT-Small-patch16-224* and *DeiT-Tiny-patch16-224*. As the model names indicate, all models operate on images of size 224×224 using a patch size of 16. The first model ViT-Base-patch16-224 was pre-trained on ImageNet-21k[1] (14 million images, 21,843 classes) at resolution 224x224, and fine-tuned on ImageNet-1k[7] (1 million images, 1,000 classes) at resolution 224x224. The DeiT variants were pre-trained on ImageNet-1k only, but their training is based on distillation. We downloaded pre-trained models from Huggingface (https://huggingface.co/models).

3 TASK

We describe the image classification task in detail in the following subsections.

3.1 Object classification

In this classification task, we consider classification of sonar image snippets that are extracted around locations proposed by an object detector. A given sonar image snippet is classified into 4 classes: cylinder, truncated cone, wedge and clutter. The first three classes represent regular geometric shapes whereas the last one is a composite class containing spurious detections and other objects that do not belong to the first three classes.

Figure 2 illustrates the four-class classification task considered in which the input to the classifier is a sonar image and the output is the probabilities of the classes considered.

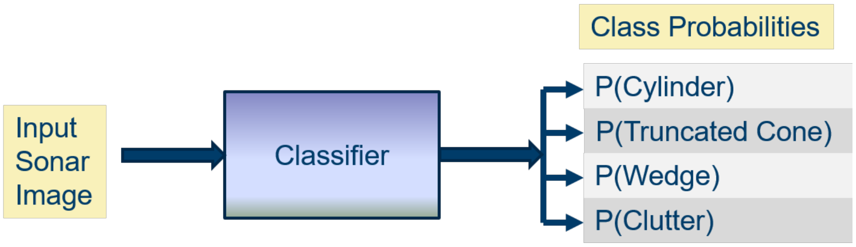


Figure 2: Classification Problem

We implemented a baseline system that realizes the task, by adding a classification head to a base-CNN. In our implementation, we used the Inception-Resnet-V2 architecture[8] as the base-CNN and a single layer fully connected neural network as the classification head.

Then several Vision Transformers were applied to the same task. More specifically we used ViT-Base-patch16-224[2], deit-tiny-patch16-224, deit-small-patch16-224 and deit-base-patch16-224[9] as the classifier.

3.1.1 Data set

The data set used in this work consists of Synthetic Aperture Sonar (SAS) images collected using a HISAS 1030 sensor[3] mounted on a HUGIN autonomous underwater vehicle[†]. This is a data set where ground truth labels (i.e. class labels of interesting objects) can easily be obtained because locations of the objects of regular shapes (cylinder, truncated cone and wedge) are known. The sonar images were first sent through a blob detector and image snippets of size 299x299 pixels were extracted around each detection. In this way about 90000 snippets were collected, where a vast majority of the images belonged to the clutter class. More specifically, there were about 5000 images of cylinder, truncated cone and wedge objects whereas about 84000 images contained clutter objects. This is clearly a highly unbalanced data set and therefore class weighting was applied during training to counter this imbalance. The dataset set was augmented through flipping along the across-track direction and random translations. About 90% of the total images were used as the training set and the remaining images were set aside as the test set. This resulted in a final training and test set sizes of 226000 and 32000 images respectively. Note that we do not use a validation set.

3.1.2 Training and Evaluation

In the case of CNN classifier, the base-CNN was initialized with the original, pre-trained parameter values, whereas the classification network is initialized with random values. We followed two main training strategies, full fine-tuning and partial fine-tuning. In full fine-tuning, the whole network was trained using the training set described above, whereas in partial fine-tuning, only the classification head was trained. We employed the update rule of stochastic gradient descent (SGD) with momentum together with a batch-size of 25 images. In each experiment, the system was trained for 25 epochs.

The loss function used in training of the baseline system is the categorical cross entropy (CE). That is

[†] https://www.kongsberg.com/discovery/autonomous-and-uncrewed-solutions/hugin/

$$L_{\text{baseline}} = -\frac{1}{N}\sum_{i=1}^{N}\log P(C_{t_i}),$$

where t_i is the target class of the i^{th} sample, $P(C_j)$ is the probability of object class j and N is the number of samples in the training set.

Similar to the case of CNN classifiers, we initialize the ViT classifiers with their pre-trained parameter values except for the classification heads which are initialized with random values. Finally, full fine-tuning was performed on all the ViT classifiers, while partial fine-tuning (i.e training only classification head) was performed on the ViT-Base-patch16-224 architecture.

After each training epoch, the performance of the classifier considered is evaluated on the test set.

We calculated several evaluation metrics on the test set after each training epoch.

- **Accuracy:** This is the ratio between the number of correctly classified images and the total number of images in the test set. This is a metric not suitable for a highly imbalanced data set like ours.
- **Average Recall:** We calculated Recall averaged over all classes, i.e.

  $$R_{\text{ave}} = \frac{1}{C}\sum_{i=1}^{C}\frac{n_{ii}}{\sum_{j=1}^{C} n_{ij}},$$

  where n_{ij} is the number of class i images classified into class j and C is the number of classes.
- **Average Precision:** We calculated Precision averaged over all classes. i.e.

  $$P_{\text{ave}} = \frac{1}{C}\sum_{i=1}^{C}\frac{n_{ii}}{\sum_{k=1}^{C} n_{ki}},$$

  where n_{ij} is the number of class i images classified into class j and C is the number of classes.
- **Average F1-score:** F1-score combines precision and recall. Our estimation of F1-score averaged over all classes is based on the formula:

  $$F1_{\text{ave}} = \frac{1}{C}\sum_{i=1}^{C}\frac{n_{ii}}{\sum_{j=1}^{C}\sum_{k=1}^{C}(2n_{ii} + n_{ij} + n_{ki})},$$

- **Average Area Under the Curve:** Unlike the previous two, this metric does not depend on a particular threshold in classification. Therefore, this is a more suitable metric for our problem. We first create receiver operating characteristics (ROC) curves, that is the graph of the true positive rate against the false positive rate, for each of the classes. Then the area under the ROC curve (AUC) is calculated for each class and the final metric is obtained by averaging AUC values for all classes.

4 EXPERIMENTS AND RESULTS

We conducted several experiments to compare the performance of CNN classifier with the ViT based classifiers.

The first experiment deals with partial fine-tuning of the classification networks, i.e. training only the classification head while keeping the parameters of the base network frozen at the pre-trained values on the optical images of the ImageNet dataset. Table 1 shows the performance metrics for the CNN-based and ViT-based architectures, *Inception-Resnet-V2* and *ViT-Base-patch16-224*.

Table 1: Evaluation metrics for partial fine-tuning.

Network	fea-vec-dim	Accu	Prec	Rec	F1	AUC
Inception-Resnet-V2	1536	95.65	58.67	39.35	44.78	**93.12**
ViT-Base-patch16-224	768	**96.00**	**64.28**	**41.32**	**47.64**	93.00

Partial fine-tuning measures the ability of the pre-trained network to generate rich enough features for the classification of SAS images. Both the CNN and ViT could generate sensible features that lead to decent accuracy and AUC values. However, recall and F1-score are poor for both of the networks. Evaluation metric values for the CNN are very similar to those of the ViT, making it hard to decide which one is better. However, the feature vector dimension of the CNN is almost twice as large as that of the ViT. That means that ViT features can be considered to be a more compact representation of the same amount of information.

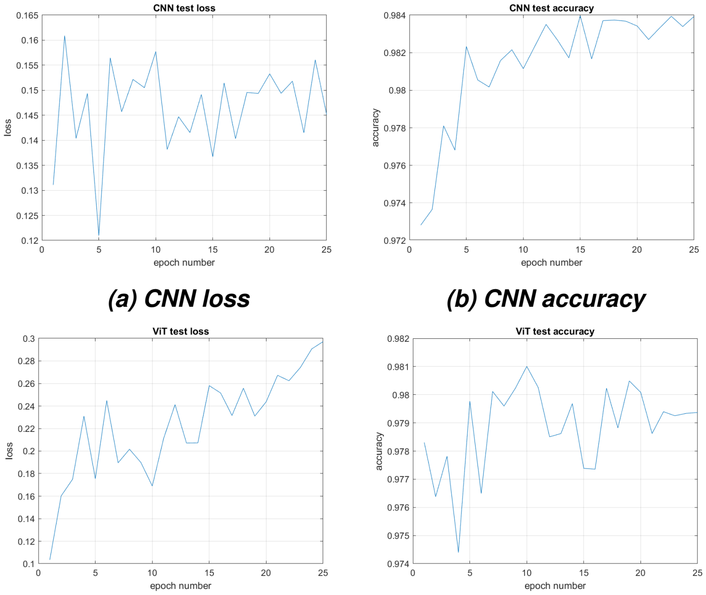
In the second set of experiments full fine-tuning was performed on both the CNN based and ViT based classification networks. The results are shown in Table 2.

Table 2: Evaluation metrics for full fine-tuning.

Network	#Parm	Accuracy (ImageNet)	Accu	Prec	Rec	F1	AUC
Inception-Resnet-V2	55.8M	80.1	**98.66**	97.41	**87.33**	**91.77**	98.09
ViT-Base-patch16-224	86M	88.6	98.36	95.31	85.70	90.03	97.57
DeiT-Tiny-patch16-224	6M	76.6	98.27	95.64	81.99	87.94	97.97
DeiT-Small-patch16-224	22M	82.6	98.34	**98.44**	80.79	88.18	**98.72**
DeiT-Base-patch16-224	87M	84.2	98.37	97.36	83.62	89.37	97.75

In this case, we have used three variants of the DeiT model[9] in addition to ViT-Base-patch16-224 and Inception-Resnet-V2. Comparing Tabel 2 with Table 1, it is clear that full fine-tuning leads to better performance metrics than partial fine-tuning. This may be due to the fact that sonar images and optical images have considerably different characteristics. Focusing only on the fully fine-tuned models in Table 2, it is apparent that the accuracy is almost the same across all models. Inception-Resnet-V2 has the best Recall and F1-score, whereas the DeiT-Small-patch16-224 has the best Precision and AUC values.

If we focus on the metrics AUC and accuracy, there is no considerable difference in performance among the different models. This happens to be the case, even though there is a considerable variance of the accuracy of the original models on the optical images from the ImageNet. A possible explanation is that factors such as model size (number of parameters) can even out the superiority of some original models. For example, ViT-Base-patch16-224 is far better in accuracy for optical images than Inception-Resnet-V2, but it has much higher number of parameters than its CNN counterpart. Therefore, in fine-tuning with a smaller sonar dataset, ViT-Base-patch16-224 cannot maintain its generalization ability as much as Inception-Resnet-V2 does, and hence losing its superiority. On the other hand, DeiT-Tiny-patch16-224 has the lowest number of parameters and hence it can achieve a good generalization ability on the smaller sonar data set, countering its low accuracy on optical images. DeiT-Small-patch16-224 which has 22 million parameters seems to be good balance between the number of parameters and original accuracy, as it turned out to be the best ViT model.



(a) CNN loss ***(b) CNN accuracy***

(c) ViT loss ***(d) ViT accuracy***

Figure 3: Development of test metrics during training for CNN (Inception-ResNet-v2) and ViT (Vit-base-patch16-224).

Figure 3 shows the development of the loss and accuracy of the test set after each epoch of training (fine-tuning). In this case, we have considered only Inception-Resnet-v2 and ViT-Base-patch16-224 representing CNNs and ViTs respectively. As can be observed from the figure, development of loss for both the CNN and ViT appears to be erratic. Ideally the loss should go down as training progresses, but this behaviour may indicate a mismatch between training set and test set distributions. Development of accuracy, however, shows an interesting behaviour. In the case of the CNN, accuracy grows throughout the training period, whereas in the case of ViT, accuracy reaches a peak and then starts to drop. This indicates that the ViT can overfit the sonar dataset more quickly than the CNN, and this agrees with the fact that ViT-Base-patch16-224 has a considerably higher number of parameters than Inception-Resnet-v2.

5 CONCLUSION

Vision Transformers considered in this work can successfully be fine-tuned on SAS data for classification of objects in sonar images. The resulting ViTs can give performance comparable to that of the CNN architecture considered. Even though big ViTs have superior performance over CNNs on optical images, it is difficult to achieve the same superiority by fine-tuning them on our dataset which is relatively small and contains a kind of images different to optical images. A more promising approach would be to select relatively smaller, yet reasonably well performing ViTs such as ViT-Small-patch16-224 for fine-tuning on smaller sonar datasets.

ACKNOWLEDGMENTS

We thank the Royal Norwegian Navy, Kongsberg Discovery for providing HISAS data used in the work. We also thank the NATO STO Centre for Maritime Research and Experimentation for providing the MUSCLE SAS data, which was collected with funding from the NATO ACT.

REFERENCES

1. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi:10.1109/CVPR.2009.5206848.
2. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020, 2010.11929. URL `https://arxiv.org/abs/2010.11929`.
3. P. E. Hagen, T. G. Fossum, and R. E. Hansen. HISAS 1030: The next generation mine hunting sonar for AUVs. In *UDT Pacific 2008 Conference Proceedings*, 2008.
4. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 2012. doi:10.1145/3065386.
5. C. Li, Z. Huang, J. Xu, and Y. Yan. Underwater target classification using deep learning. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1–5, 2018.
6. J. Maurício, I. Domingues, and J. Bernardino. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, 13(9), 2023. doi:10.3390/app13095521.
7. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015. doi:10.1007/s11263-015-0816-y.
8. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016, 1602.07261.
9. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020, 2012.12877. URL `https://arxiv.org/abs/2012.12877`.
10. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.
11. N. Warakagoda and Ø. Midtgaard. Fine-tuning vs full training of deep neural networks for seafloor mine recognition in sonar images. In *Underwater Acoustics Conference and Exhibition (UACE), Skiathos, Greece*, 09 2017.
12. D. P. Williams. Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2497–2502, 2016.