Visualizing and Interpreting Transformer-based Vision Models

Jorge Roldan JLR9718@NYU.EDU
Courant Institute of Mathematical Sciences

Timothy Xu TIMOTHY.XU@MATH.NYU.EDU
Courant Institute of Mathematical Sciences

Juexiao Zhang JZ4725@NYU.EDU
Courant Institute of Mathematical Sciences

Editor:

Abstract

Transformer-based vision models are increasingly popular and we need better ways to interpret and visualize their predictions. Previous works have been limited to visualizing attention maps; we apply a Shapley-value based method (FastSHAP) to Vision Transformers and Masked Autoencoders, comparing the results to a classical ResNet. We find that choosing ResNet as the *surrogate* model for FastSHAP lets us successfully interpret and visualize transformer-based vision models. We observe that the estimated Shapley values of ResNet and ViT trained on CIFAR-10 are qualitatively different, even though the models' predictions are mostly consistent.

Keywords: Interpretability, Visualization, Shapley values, Vision Transformer, Masked Autoenconder

1. Introduction

Transformer-based vision models like ViTs (Dosovitskiy et al. (2020)) and MAEs (He et al. (2021)) have shown remarkable potential for image representation and classification. Previous efforts to visualise and interpret the predictions given by these complex deep neural networks have been limited to visualising the 'attention maps' in the transformers. Whilst in general there have been numerous proposed methods to interpret and visualise complex machine learning models (Covert et al. (2020)), most of them rely on intuitions and educated guesses and lack solid mathematical grounding (Hooker et al. (2018)); in fact it is unclear whether some of the methods actually work at all (Jethani et al. (2021a)). We implement a proven visualisation and interpretation method based on Shapley values, which originate in the field of cooperative game theory. The fundamental difficulty with applying Shapley values to high-dimensional problems is the computation time; the calculations suffer from the *Curse of Dimensionality*. The ingenuity of FastSHAP, proposed by Jethani et al. (2021b), was to use a deep *surrogate* model to approximate how the model we want to interpret will classify inputs with some features missing (masked), and another deep *explainer* model to estimate the Shapley values for different pairs of inputs and predictions (without the need for any ground truth Shapley value data for training). We apply FastSHAP.

2. Related Works

2.1 Transformers

Since it was proposed in Vaswani et al. (2017), the transformer model has shown its effectiveness in different fields of deep learning applications. Following the success of transformer models in natural language processing such as BERT (Devlin et al. (2018)), people have been exploring the adapataion of transformers to various computer vision tasks. DETR proposed by Carion et al. (2020) is the first work that achieved state-of-the-art level performance with transformers in the task of object detection. As for the famous ImageNet classification task which was previously dominated

by convolutional neural networks such as ResNet (He et al. (2015)), transformers also have achieved competitive performance. The Vision Tranformer (ViT) model proposed by Dosovitskiy et al. (2020) shows for the first time that without CNNs, a pure transformer architecture can perform very well on image classification. Furthermore, in the unsupervised learning setting, a recently proposed Masked Autoencoder (MAE) by He et al. (2021) shows promising results; MAEs use a transformer to build an encoder-decoder architecture to learn image representations without supervision. In conclusion, much progress has been made to apply transformers to computer vision and it has achieved performance comparable to state-of-the-art CNN-based models.

2.2 Visualizations

Due to the recent proliferation of deep neural networks which have too many parameters for a human to understand, we have seen many proposed black box methods to interpret and visualize complex machine learning models, starting with the introduction of LIME (Ribeiro et al. (2016)). LIME stands for Local Interpretable Model-agnostic Explanations and does exactly that: given a trained ML model, given any input, LIME finds a linear model that locally approximates that model, only around that specific input. Shapley value-based methods were then proposed (Lundberg and Lee (2017)); these methods are provably helpful due to the mathematical properties of Shapley values. Later works put the wide gamut of interpretability methods under the spotlight: Hooker et al. (2018) highlights the fact that many previous interpretability models do not retrain their models on masked data, thus violating the machine learning maxim requiring training and evaluation data to come from the same distribution. Hooker et al. propose ROAR (RemOve And Retrain) to properly evaluate interpretability methods. Covert et al. (2020) unifies a staggering 25 existing machine learning methods including SHAP (which is based on Shapley values) and LIME. Jethani et al. (2021a) shows that the class of *joint amortized explanation methods* in fact encode predictions in their interpretation models (making the interpretations not credible), and fail to select features involved in control flow only.

Specifically for deep neural networks, Zeiler and Fergus (2014) provide wonderful visualizations to modern deep convolutional neural networks. However for transformers, previous work have been mostly limited to visualizing attention maps. Although that is indeed a choice, reasoning about how attention patterns are related to model outputs is difficult, since there can be dozens to hundreds of attention maps in a single transformer. Chefer et al. (2021) proposed another way to interpret transformers. We think that given the increasing prevalence of transformer models in computer vision, exploring more ways to visualize and interpret them should be a priority. Kokalj et al. (2021) applied SHAP to the transformer-based text classifier BERT (Devlin et al. (2018)); prior to our work, there had been no applications of SHAP to transformer-based image classifiers. Indeed before FastSHAP (Jethani et al. (2021b)), high-dimensionality of images made Shapley value computations infeasible. FastSHAP provides a way around this *Curse of Dimensionality*, allowing us to apply SHAP to Transformer-based vision models like ViT and MAEs.

3. Methods

3.1 Shapley Values

Shapley values are provably useful for intepreting complex machine learning models due to their mathematical properties (Rozemberczki et al. (2022)). To explain a machine learning model's predictions, we want to identify the relative importance of each input feature, given a fixed input x_0 and the model's prediction: which features contributed the most to this model making this prediction, and which features negatively contributed to (shunned this model away from) the prediction it made? This can be done by estimating the model's prediction on x_0 if some features are missing (masked). The Shapley value of a particular feature value is the **average marginal contribution**

of that feature value across all possible combinations of feature presence, for that fixed input x_0:

$$\phi_i(v) = \frac{1}{d} \sum_{s_i \neq 1} \binom{d-1}{\mathbf{1}^\top s}^{-1} (v(s + e_i) - v(s))$$

Here i enumerates over features, v is a function from $2^d \mapsto \mathbb{R}$ mapping each choice of combinations of feature presence/masking to its expected output by the model (which can be the probability it assigns to a certain fixed class), and $s \in \{0,1\}^d$ encodes which features are present (not masked). By above formula, we see that the Shapley value of a feature value is a weighted average of that individual feature value's contribution to the prediction, across all possible combinations of *masking* the different features of this input. The specific weights are chosen precisely such that the Shapley values obey the mathematical properties of *Efficiency, Symmetry, Dummy,* and *Additivity*; in fact Shapley values are the unique values which obey these 4 properties (Shapley (2016)).

Efficiency: the Shapley values over all features of a certain input x_0 sum to the difference between the model's prediction for x_0 and the average of the model's predictions over all x.

Symmetry: for any input x_0, the Shapley values of 2 feature values in x_0 are the same if these 2 feature values contribute equally to the model's prediction on all possible combinations of feature masking on x_0.

Dummy: for any input x_0, if a feature value in x_0 has no (positive or negative) contribution to any of the model's prediction on any possible combinations of feature masking on x_0, then its Shapley value is 0.

Additivity: If a model c always predicts the sum of predictions of model a and model b, then the Shapley values of any feature value in any input x_0 for model c should be the sum of the Shapley values for the same feature value and input for models a and b.

3.2 FastSHAP

We use FastSHAP as introduced in Jethani et al. (2021b). For proofs of all assertions in this section please see the original paper. Instead of calculating Shapley values using the original weighted sums formulation, FastSHAP uses a deep model to estimate Shapley values by minimising a certain loss function, which (up to an *efficiency gap* $\frac{1}{d}\left(v_{x,y}(\mathbf{1}) - v_{x,y}(\mathbf{0}) - 1^\top \phi_{\text{fast}}(x, y; \theta)\right)$, which can either be added explicitly, or implicitly normalised via penalisation during training) turns out to also characterize Shapley values:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{p(\mathbf{x})} \mathop{\mathbb{E}}_{\text{Unif}(\mathbf{y})} \mathop{\mathbb{E}}_{p(\mathbf{s})} \left[\left(v_{\mathbf{x},\mathbf{y}}(\mathbf{s}) - v_{\mathbf{x},\mathbf{y}}(\mathbf{0}) - \mathbf{s}^\top \phi_{\text{fast}}(\mathbf{x}, \mathbf{y}; \theta) \right)^2 \right].$$

For empirical considerations regarding gradient descent, various techniques such as mini-batching are used to reduce gradient variance for faster training of the FastSHAP explainer.

In order to approximate the model's expected prediction on inputs with masked features, instead of marginalising out the masked features explicitly which will be computationally infeasible, FastSHAP proposes using a supervised deep *surrogate* model. The surrogate model takes as input a vector of masked features $m(x, s)$ where the masking function m replaces features x_i such that $s_i = 0$ with a mask value that is not in the support of any input data. This surrogate model is trained to minimise the loss function (based on KL divergence) below:

$$\mathcal{L}(\beta) = \mathop{\mathbb{E}}_{p(\mathbf{x})} \mathop{\mathbb{E}}_{p(\mathbf{s})} \left[D_{\text{KL}}\left(f(\mathbf{x}; \eta) \parallel p_{\text{surr}}(\mathbf{y} \mid m(\mathbf{x}, \mathbf{s}); \beta) \right) \right].$$

It has been proven that the global minimizer to the loss equation above is indeed the expected prediction function on inputs with masked features.

4. Experiments

4.1 Experimental Setup

We have two groups of experiments: comparing performance of various surrogate models, and comparing the FastSHAP visualizations of transformer-based models to classical ResNet. We conduct our experiments on the CIFAR-10 dataset due to computational limits following the scripts of Jethani et al. (2021b), but we believe our observations can be extended to larger datasets and more general cases. Following the conventions in Jethani et al. (2021b), we refer to the vision models as the *original models*, the surrogate models approximating the original models as *surrogates*, and the models trained to estimate the Shapley values as the *explainers*. We use the same UNet architecture as in Jethani et al. (2021b) for explainers in all experiments.

4.2 Models and Surrogates

Originally, ResNet, ViT, and MAE were mainly applied to larger data sets such as ImageNet. We trained each of them from scratch on CIFAR-10; details are in the appendix. Note: we are abusing the term *MAE* here; see appendix for details. All 3 models achieved similarly high levels of accuracy at around 90%.

In FastSHAP, the surrogates accept inputs with randomly masked-out features and try to produce predictions similar to the original model, as measured by the KL divergence. In Jethani et al. (2021b), the authors used ResNet18 as the surrogate for the original ResNet model. This is a reasonable choice because we want the surrogate to approximate the original model. To explore other choices we trained and tested both ResNet18 and ViT surrogates on all 3 original models. The ViT surrogate was chosen to have the same exact architecture as the original ViT we trained. Since the MAE's encoder uses a ViT with different dimensions, we also trained a surrogate using that different ViT for a fair comparison. The two ViTs are denoted "ViT(384+384)" and "ViT(196+384)" respectively according to their dimensions. We compared all the surrogates on the validation data set with a complete set of features (no masking). The validation loss (as measured by cross entropy), validation accuracy, and the KL divergence loss (during training) between the surrogates and the original models are reported in table 1.

4.3 Visualizations

After training original models, surrogates, and corresponding explainers, we compute and visualize FastSHAP values for pairs of input images and output classes. We designed 3 visualization tasks:

1. Visualize original models. Using randomly selected images, for each model, we compute the prediction probabilities and visualise the FastSHAP values over all possible classes. The appendix shows examples of visualizing ResNet and ViT using 10 randomly selected images.
2. Visualize and contrast between pairs of original models, in relation to the models' predictions. Based on the predictions of original models A and B, we partition the validation data set into 4 subsets: "A wrong B wrong" (meaning both A and B made wrong predictions), "A wrong B correct", "A correct B wrong" and "A correct B correct". We aggregate the statistics of the 4 subsets (table 3 in appendix) and conduct visualizations for images in each subset separately.
3. Compare surrogate models' behavior. We try different surrogate models for the same original model, and conduct visualizations using explainers trained on the different surrogates.

In all visualizations below, blue and red pixels mean negative and positive Shapley values respectively; intensity corresponds to magnitude. In figures 1 and 2, models A and B are ResNet and ViT respectively. The first rows are visualizations from ResNet with ResNet surrogates; second rows are ViT with ViT surrogates; third rows are ViT with ResNet surrogates. More visualizations (including the other 2 subsets) are in the appendix. Note: during explainer trainings, ViT with a ResNet surrogate had much lower losses (different order of magnitude) than all other pairs of choices.

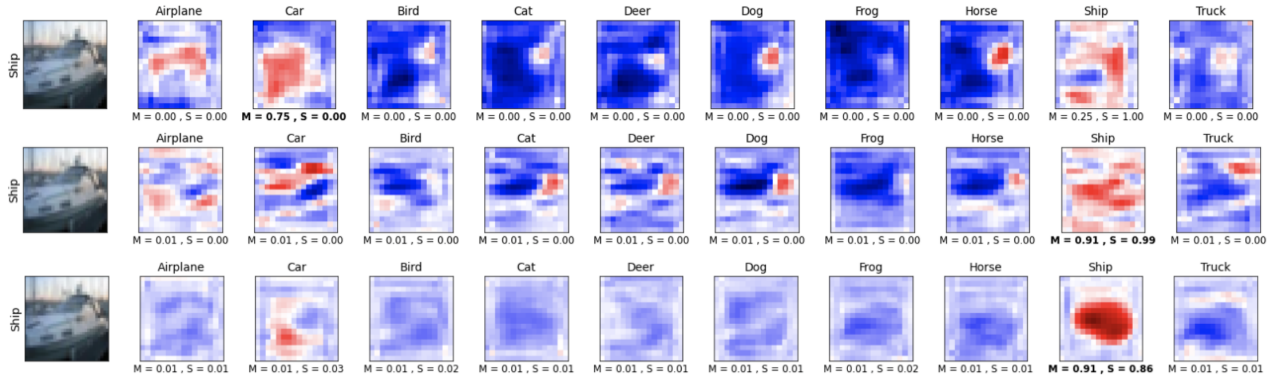

Figure 1: A Wrong B Correct

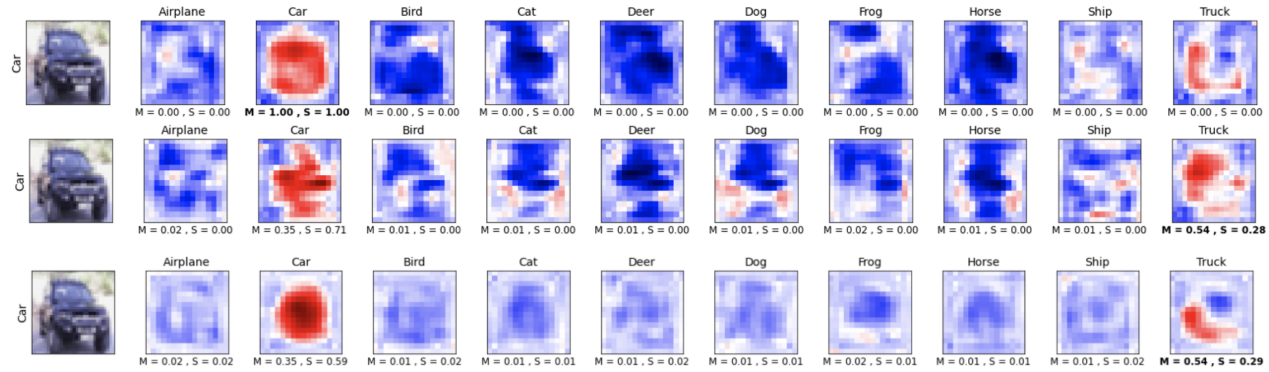

Figure 2: A Correct B Wrong

5. Discussion

Table 1: Surrogate Models Results

Original by Surrogate	**Val Loss**	**Val Accurracy**	**KLDivLoss**
ResNet by ResNet	0.4414	0.8860	0.4045
ResNet by ViT	1.685	0.7744	1.6529
ViT by ViT	0.7980	0.7531	0.4993
ViT by ResNet	0.3364	0.9195	0.1973
MAE by ViT(384+384)	0.8457	0.7289	0.7324
MAE by ViT(196+768)	0.8048	0.7251	0.6972
MAE by ResNet	0.5526	0.8912	0.7124

1. Per table 1, ResNet18 works better as a surrogate model in all cases, even when the original model has a completely different architecture. In fact when applied to ViT it performs even better than the original model! This is likely due to ResNet18 already having the best performance among the 3 original models. Dosovitskiy et al. (2020) indicates that ViT scales well with a large amount of parameters, so perhaps a larger ViT will be a better surrogate. We leave this to future work.

2. In table 1, we see that small KL divergence loss during the training of the surrogate is predicative of small cross entropy validation loss (on non-masked inputs), but it cannot predict a high validation accuracy (on the same non-masked inputs) when contrasted with the original model. For example approximating ResNet by a ResNet surrogate model has 0.4045 KLDivLoss with the surrogate working well at 88.6% accuracy, yet approximating ViT by ViT surrogate model has similar small KLDivLoss but validation accuracy drops sharply to 75.3% (even though the original ViT model achieved 90% accuracy).

3. To improve surrogate training, one possible future work is to choose a different loss from KLDiv. Reverse KLDivLoss may be a good choice since we are only interested in the argmax of the outputs for classification, and we know Reverse KLDivLoss will approximate a smaller area better. With this modification ViT may work better as surrogates.

4. We did not experiment with architectures other than UNet for the FastSHAP explainer and this would be a possible route of future research. Evaluating the FastSHAP outputs is still an open question; we do not have the ground truth Shapley values for images hence we lack an objective benchmark giving the accuracy of our Shapley value estimates. We can manually view the visualizations showing the Shapley value estimates, and inspect the explainer training losses (which as we have discussed above, may not translate directly to faithfulness of predictions).

5. Our paper gives more evidence to the wide applicability of FastSHAP as a flexible black box interpretability method.

6. Per table 3 in appendix, over 90% of ResNet and ViT's predictions are consistent. This seems to suggest ResNet and ViT behave similarly for image classification despite their big differences. This could offer support to a recent work by Liu et al. (2022) where a slightly-modified ResNet (inspired by ViT) had increased performance, to a level similar to ViT.

7. Per figures 1, 2, 3, 4, we see that whilst ResNet using ResNet surrogate and ViT using ViT surrogate produced qualitatively similar visualisaitons, ViT using ResNet surrogate (third rows) seemed to produce qualitatively different estimated Shapley values. Since ResNet performed better as surrogates and this particular pair choice also had much lower explainer training losses, we have more confidence that the third rows reflect the truth Shapley values better. Contrasting first and third rows, we may deduce that ResNet and ViT assign different importances to features (even though per point above, the classification predictions are mostly consistent). ViT seems to tend to select an entire central patch as important, perhaps reflecting the attention mechanisms within transformers. Future experiments may try to investigate this further, and test whether this is because images in CIFAR-10 are all centered.

8. Per same figures, comparing second and third rows we see the importance of surrogate choice in FastSHAP: both visualize ViTs yet the outputs seem qualitatively different. Again both surrogate validation data and explainer training losses suggest the third rows are more accurate.

9. Per the same figures, we did not find any interesting differences between the visualisations for the 4 different subsets partitioned by model predictions.

6. Conclusions

The choice of a surrogate is important for FastSHAP, with ResNet beating ViT. Interestingly, a small KLDivLoss when training a surrogate for a high-performing original model is predicative of small CrossEntropyLoss when validating the surrogate on non-masked inputs, but neither can predict a high-performing surrogate. Overall, Shapley value-based methods can be applied to interpret transformer vision models well, using FastSHAP. We see Shapley values of ResNet and ViT trained on CIFAR-10 are qualitatively different, even though the predictions are mostly consistent.

References

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, June 2021.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation, 2020. URL `https://arxiv.org/abs/2011.14878`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks, 2018. URL `https://arxiv.org/abs/1806.10758`.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. 2021a. doi: 10.48550/ARXIV.2103.01890. URL `https://arxiv.org/abs/2103.01890`.

Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation, 2021b. URL `https://arxiv.org/abs/2107.07436`.

Enja Kokalj, Blaž Škrlj, Nada Lavrač, Senja Pollak, and Marko Robnik-Šikonja. BERT meets shapley: Extending SHAP explanations to transformer-based classifiers. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 16–21, Online, April 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.hackashop-1.3`.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 2017. doi: 10.48550/ARXIV.1705.07874. URL `https://arxiv.org/abs/1705.07874`.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. URL `https://arxiv.org/abs/1602.04938`.

Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning, 2022. URL `https://arxiv.org/abs/2202.05594`.

L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, 2016. doi: doi:10.1515/9781400881970-018. URL https://doi.org/10.1515/9781400881970-018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Appendix

All of our code and notebooks can be found in our GitHub repository [1]. Original model training details:

ResNet18: we followed the script and hyperparameters in Jethani et al. (2021b) to train a ResNet18 model on CIFAR-10 and obtained 93.25% accuracy on the validation set.

ViT: the original work by Dosovitskiy et al. (2020) trained ViT on ImageNet and larger data sets. We drew inspiration from the community[2] and trained a lighter version with roughly 6.3 million parameters from scratch. Label smoothing and auto-policy augmentation were applied to improve the performance. The model achieved 90.60% validation accuracy.

MAE: whilst the original MAE is scalable to learn large amounts of training data, we managed to make it work on a small data set, CIFAR-10, following the work of a community implementation[3]. The model is pre-trained on CIFAR-10 training data, and the encoder was taken and fine-tuned on the same set for image classification. Henceforth, this ViT classifier pretrained from MAE will simply be referred to as MAE. We obtained over 90% accuracy, similar to the result of ViT.

Table 2: Original Models Validation Loss and Accuracy

Model	**Val Loss**	**Val Accurracy**
ResNet18	0.4417	0.9325
ViT	0.3758	0.9060
MAE-pretrained ViT	0.5287	0.9091

Table 3: Subset Stats: ResNet18 (A) and ViT (B) predictions; Wrong = w, Correct = c

Subset	**Aw Bw**	**Aw Bc**	**Ac Bw**	**Ac Bc**
Size	347	328	593	8732

In figure 5, the first two rows are for task 3 (visualizing MAE) and last two rows are for task 1.

1. https://github.com/roldanjrgl/ml_project
2. https://github.com/omihub777/ViT-CIFAR
3. https://github.com/IcarusWizard/MAE

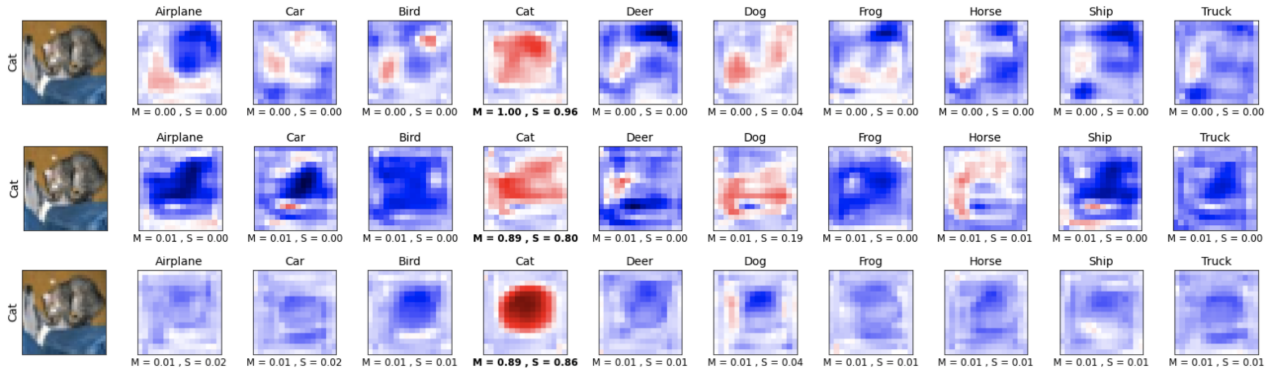

Figure 3: A Correct B Correct

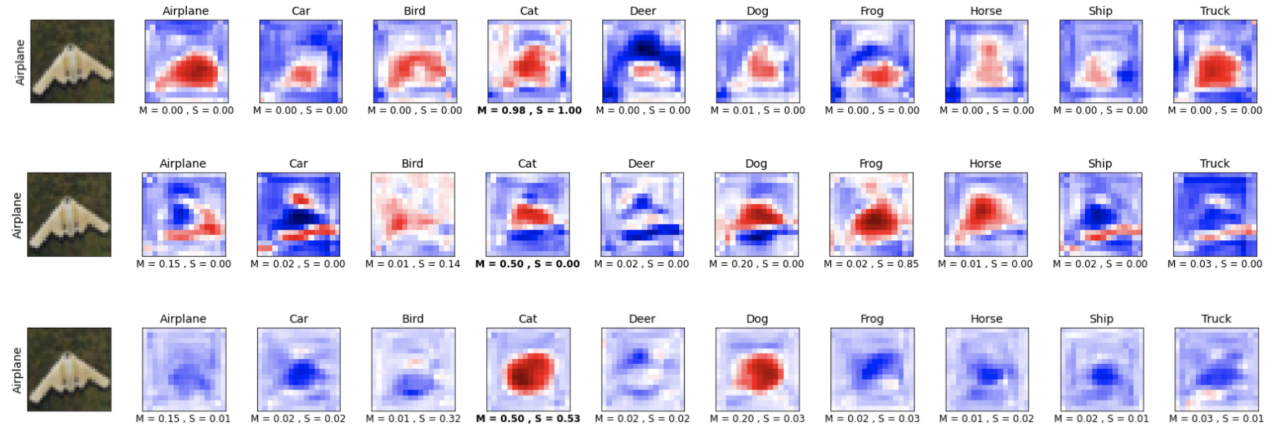

Figure 4: A Wrong B Wrong

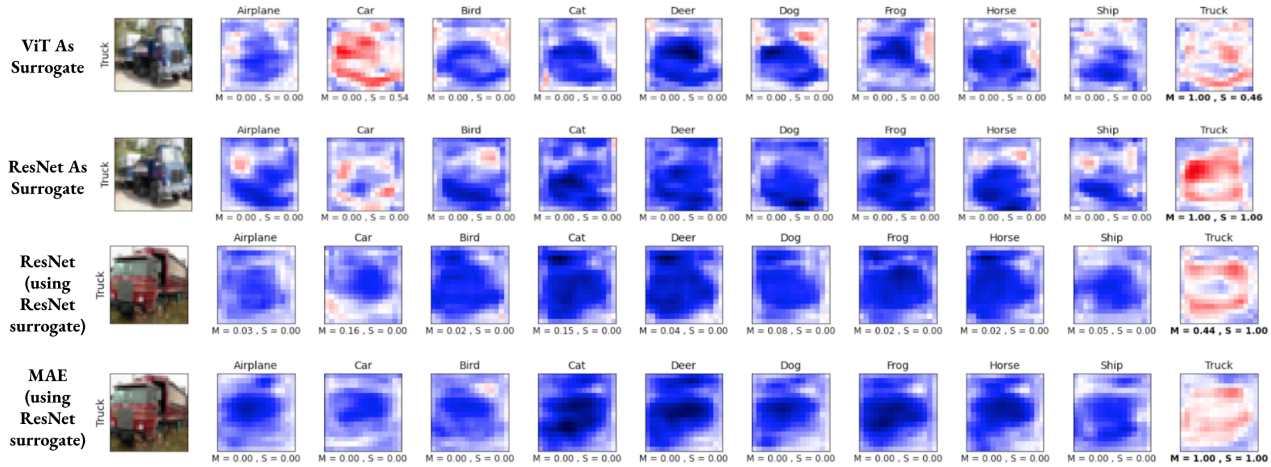

Figure 5: Visualization Examples for Tasks 1 and 3

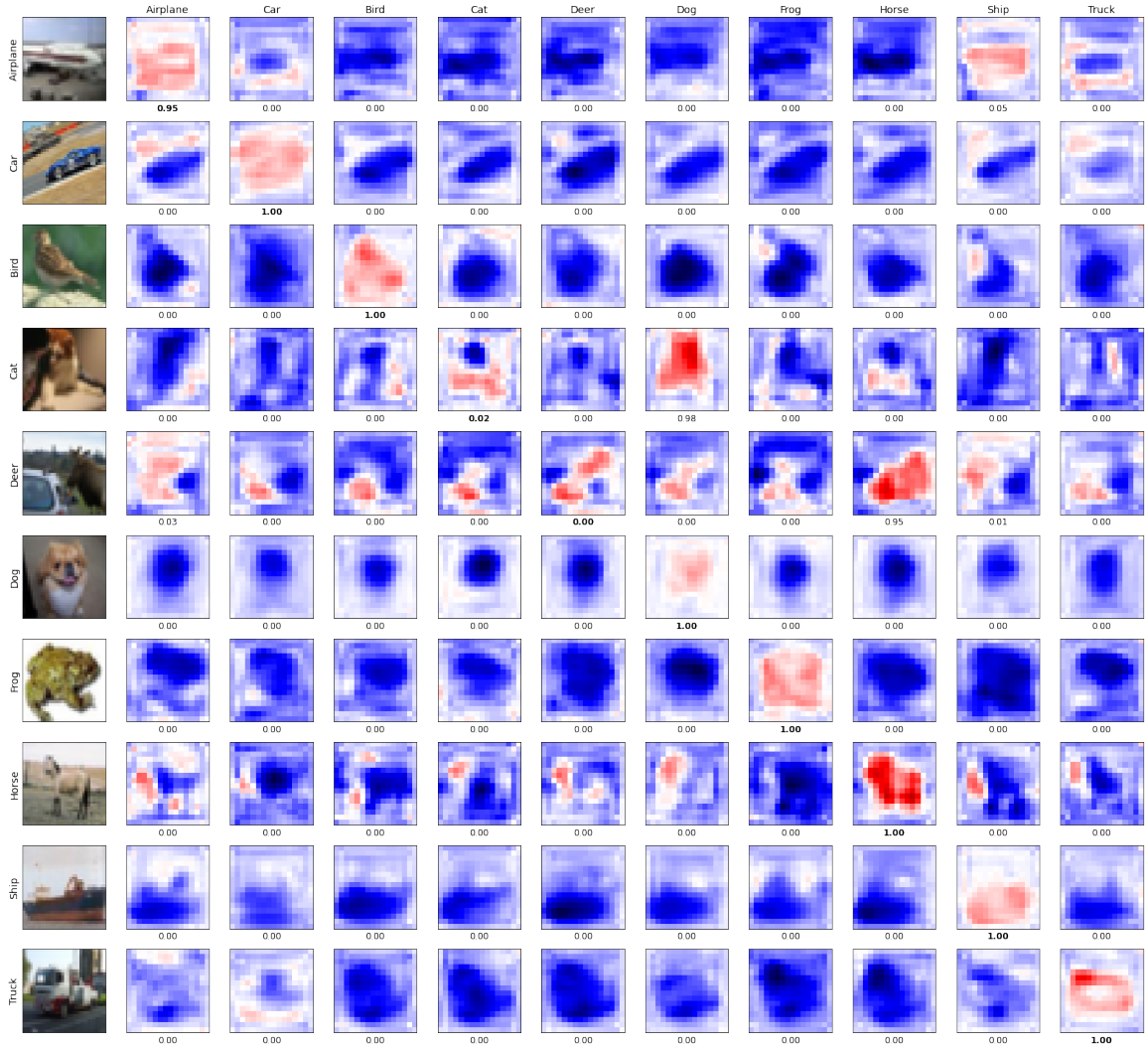

Figure 6: Visualizing ResNet18 with ResNet18 Surrogate, 10 Images Against Each Class

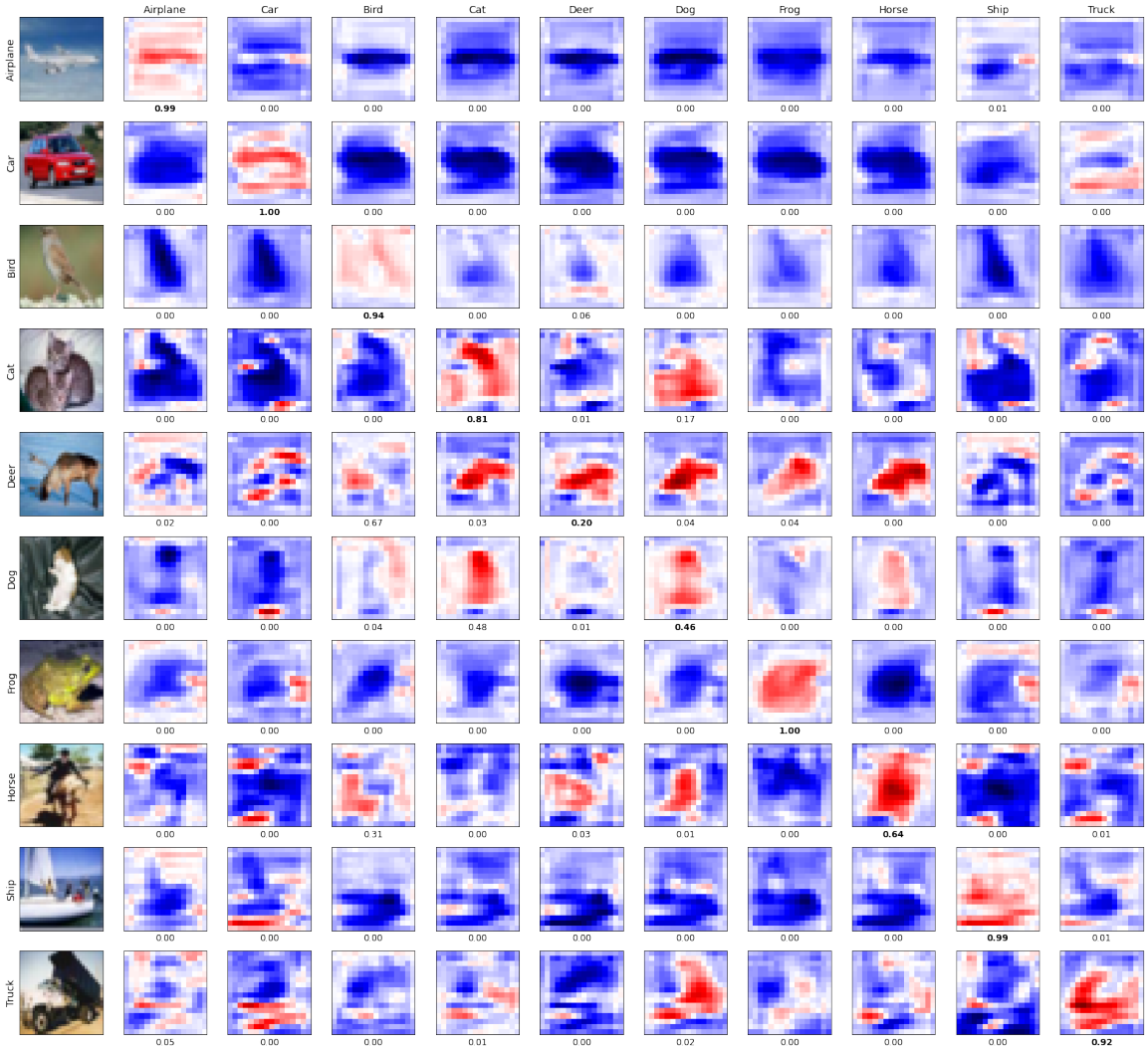

Figure 7: Visualizing ViT with ViT Surrogate, 10 Images Against Each Class