TECHNISCHE UNIVERSITÄT
ILMENAU

Department of Computer Science and Automation
Data-intensive Systems and Visualization Group

Research Project Report

Learn 2 Look: Improving Image Classification by Spatial Loss Functions

Registration Date:	15. October 2023
Submission Date:	01. March 2024
Academic Chair:	Prof. Dr.-Ing. Patrick Mäder
Academic Supervisor:	Dr. Marco Seeland
Submitted by:	Uday Kaipa born 24.10.1997 in Andhra pradesh, India uday-kumar-reddy.kaipa@tu-ilmenau.de
Course of Studies:	Research in computer and system engineering

Acknowledgments

I am immensely grateful to Dr. Rer. Nat. Marco Seeland for his support from the initial phases to the end of this project.

The project Learn2Look is one of my most valuable career experiences; it gave me immense pleasure working on the creative idea. I am grateful for the opportunity to work on such a creative and innovative idea.

I deeply appreciate the efforts of Dr.Seeland in communicating the idea, encouraging my creativity, and appreciating my effort. I am thankful to him for providing ample consultation hours and support over emails, without which this project would not have been possible.

Additionally, I would like to thank Prof. Dr.-Ing. Patrick Mäder and Dr.Seeland for sharing their invaluable knowledge in the deep learning course.

Furthermore, I would like to acknowledge the authors of the deep learning literature and other educational resources that have served as foundational references for this project. Their pioneering work has provided the groundwork upon which this research was built.

Lastly, I am grateful to all individuals, whether mentioned here or not, who have supported me in various capacities throughout the development of this project. Your encouragement and assistance have been invaluable, and I am deeply appreciative of your contributions.

Abstract

Purpose: The purpose of this study is to analyze the impact of incorporating a spatial loss term into the model on the overall training process (convergence) and the accuracy of the model.

Methods: Models are trained both with and without the inclusion of a spatial loss term to investigate its effects on classification accuracy and the number of epochs required for convergence.

Results: The inclusion of spatial loss during training has led to a reduction in the number of epochs required to reach convergence. Moreover, even with a small weighting of the spatial loss term, noticeable improvements in accuracy have been observed. Additionally, the effects of the spatial loss computed by the feature map extraction at different extraction points varied.

Conclusions: Incorporating spatial loss in conjunction with classification loss can enhance the training of a model. However, it is crucial to consider adaptive weighting strategies, as discussed in this study.

Key words: Spatial loss in image classification, Image classification.

Contents

1. Introduction

Image classification models, aim to categorise the images under a predefined set of categories, termed as labels. Further, there are different types of Image classification models. Namely multi-class classification models which categorise an image exclusively with one label, multi-label classification models which assign applicable labels to the images, and more. In this research, Multi-class classification models are considered. Classification models work by learning patterns and features from input data (e.g., images) to categorize them into predefined classes or labels. They analyze the characteristics of the data and assign it to the most appropriate category based on these learned patterns. The paper "Visualizing and Understanding Convolutional Network" [ZF13] provides insights into how these patterns are learned by the model.

Continued advancements in deep learning architectures, such as convolutional neural networks (CNNs) [GBC16], have improved the performance of image classification models. Architectures like ResNet, DenseNet, EfficientNet, and Transformer-based models like ViT (Vision Transformer) have shown significant improvements in accuracy and efficiency. In this project, MobileNetV3Small, one such architecture is considered. It is known for its emphasis on efficiency and accuracy, offering a lightweight architecture with improved performance compared to other architectures, making it suitable for mobile and edge devices.

Recent advancements in image classification includes self-supervised learning and adversarial learning techniques. Self-supervised learning include solving pretext tasks before training to learn effective representation and a loss function, which is used for actual training. A Single-stage self-classifier proposed by Amrani [AKB22]. Adversarial learning focuses on building models resistant to adversarial attacks, where noise is imposed on an image that is imperceptible for humans but makes the model provide an incorrect prediction.

A beginner would start machine learning with an image classification example of cat and dog images. Image classification is very fundamental. The potential for even marginal enhancements over existing methods holds significant advantages, given the widespread reliance on image classification as a foundational pillar in various domains.

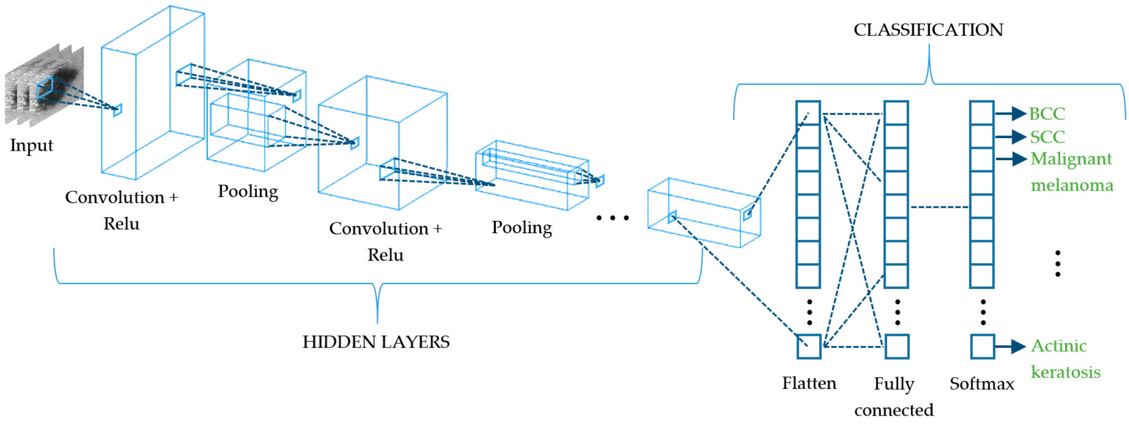

Figure 1.1. – Image classification model using CNN- [RBPGY+21]

The paper [RBPGY+21] presents an application of image classification model for skin cancer detection. Fig.1.1 describes the model architecture, which aims to detect the type of cancer. Image classification models using CNN, typically consist of Convolution layers with an activation and Pooling layers followed by flattening and fully connected layers, which are passed over a softmax activation layer.

Leveraging transfer learning techniques on pre-trained models has become a common practice to achieve high accuracy in various image-related tasks. These models are available online and integrated into frameworks like TensorFlow and PyTorch, offering architectures and pre-trained weights trained on extensive datasets like ImageNet, which contains millions of images across 1000 classes. These models excel in recognizing patterns in unseen images and can be adapted for different tasks through modification. Fine-tuning, involving retraining the pre-trained model with task-specific datasets using a smaller learning rate, facilitates the adaptation process with satisfactory accuracy.

The focal loss for Dense object detection, adds a term to the loss function that helps tackle challenges of data sets with class imbalances[LGG+17]. A similar approach is used in this project, spatial loss is added and its impact is analyzed on the training process.

Classification models commonly use a loss function like Cross-entropy loss,which measure the dissimilarity between predicted and actual class probabilities.

A novel approach that could improve the overall training process and accuracy of the model, using a spatial loss term alongside classification loss is analyzed in this study.

Spatial loss is computed through the comparison of intermediate feature maps with segmentation masks. For example, Mean Squared Error (MSE) is a widely used loss function for regression tasks, measuring the average squared difference between predicted and actual values, Which is used to compute spatial loss.

2. Methods

To evaluate the impact of including a spatial loss term, we conducted a comprehensive experiment comparing models trained with and without this term. We specifically employed the MobileNetV3Small architecture , leveraging its capabilities to classify images from the ImageNet dataset across 1000 categories. Given the distinct purposes of the datasets, we replaced the default classification head with a more intricate one to mitigate potential overfitting. Initially, we trained solely the classification head while keeping the base model frozen. Once achieving satisfactory classification performance, we proceeded to retrain the entire model.

Classification Loss: In a classification model, probabilities are assigned to each class, which are then compared to the ground truth labels to compute the classification loss. The formula for cross-entropy loss between predicted probability distribution $\hat{y}$ and true distribution y for N classes is:

$$\mathcal{L}_{\text{CE}}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

Spatial loss: It represents the disparity between the anticipated and actual positions of an object within an image, determined through metrics such as Intersection over Union (IoU) and Mean Square Error (MSE). IoU assesses the overlap between predicted and actual regions, while MSE involves a pixel-wise comparison to quantify the dissimilarity in positional information. The formula for MSE between predicted values $\hat{y}$ and true values y is:

$$\mathcal{L}_{\text{MSE}}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

The overall loss function, combining cross-entropy loss and mean squared error (MSE) with weighting factors α and β, respectively, is calculated as:

$$\text{Overall Loss} = \alpha \times \mathcal{L}_{\text{CE}} + \beta \times \mathcal{L}_{\text{MSE}}$$

In this study, different weighting methods for loss terms are evaluated and are compared with performance of the model including zero spatial loss term [β=0] i,.e without spatial loss. This experiment was iterated five times, recording metrics for each run to compute average and standard deviation values. Computing the spatial loss involves utilizing masks and intermediate feature maps. Through observation, we noted that feature maps from certain convolution blocks bear a close resemblance to the actual masks, particularly in deeper layers of the MobileNetV3Small model. Consequently, we selected three extraction points for feature maps based on the model architecture, identified as early, intermediate, and later aggregation points as depicted in fig 2.1 (with indices 61, 129, and 197 respectively).

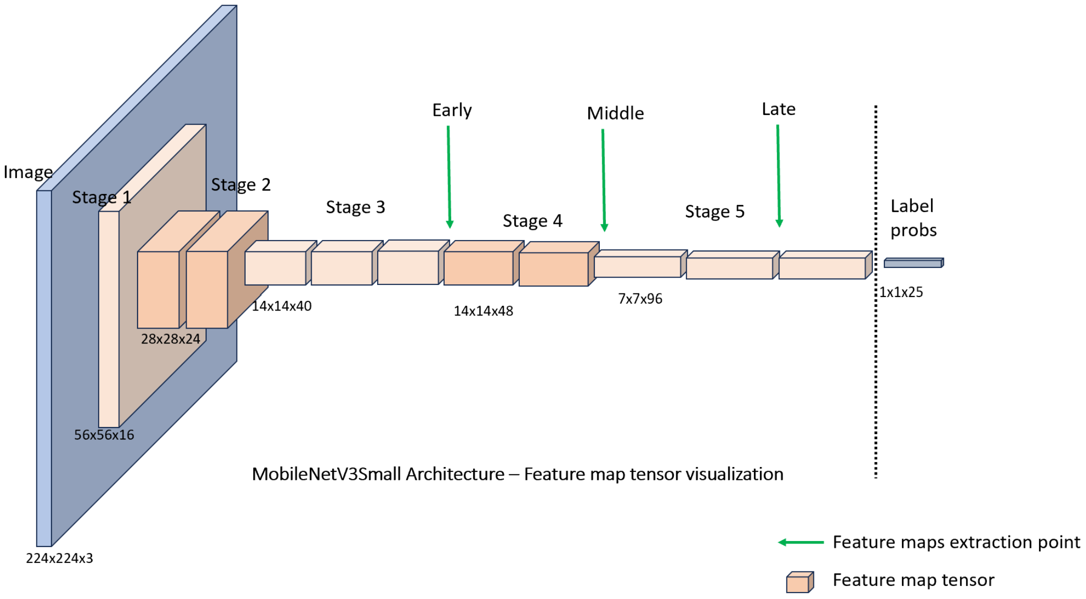

Figure 2.1. – MobileNetV3Small Feature map visualisation with extraction points

At these extraction points, we obtain feature maps as tensors, where we extract the maximum value across channels to generate single-channel feature maps. These feature maps are then aligned with the masks, ensuring sizes match for accurate comparisons. Utilizing existing loss functions in Keras, we compute the loss by contrasting the resized masks with the extracted feature maps.

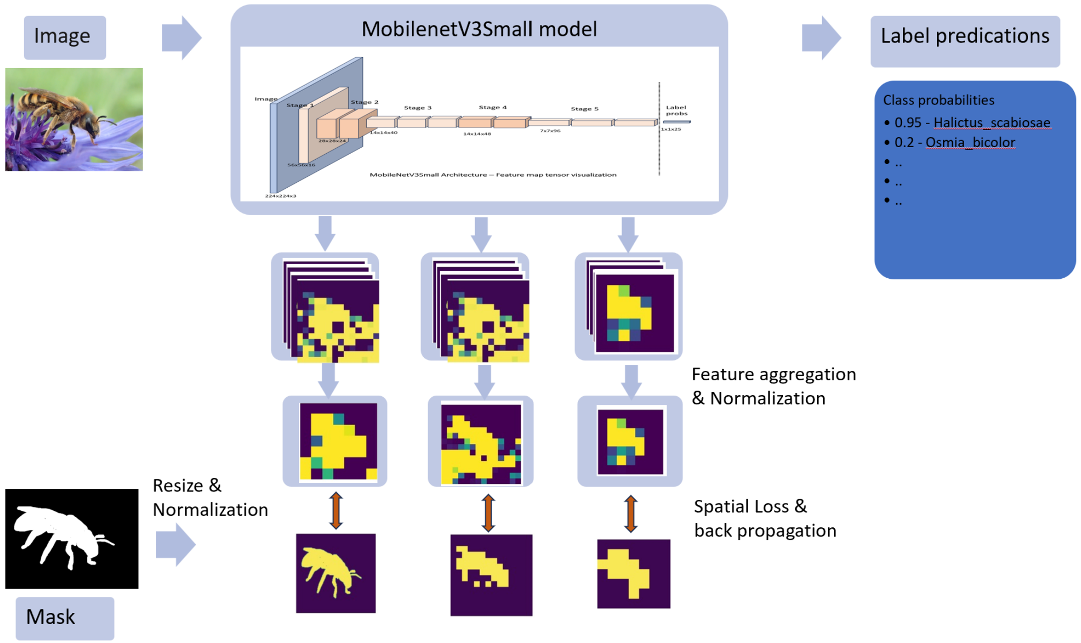

Figure 2.2. – Model Architecture

To summarize the experimental design:

1. Train the model without a spatial loss and record the metrics [β=0].

2. Train the model with the spatial loss term, experimenting with various spatial loss functions:

    - Computing spatial loss from intermediate layers' feature maps.

    - Integrating spatial loss with the classification loss and assessing performance.

Some of the additional implementations, distinguishing this model from standard image classification models, include its structure as a multiple input and multiple output model. Since the spatial loss necessitates feature maps, the sizes of these feature maps at extraction points are predetermined. Consequently, the masks loaded alongside images are resized to match these feature map sizes and are passed as outputs, representing masks at different scales. This redundancy was introduced to maintain simplicity and optimize resource efficiency.

All standard data augmentation techniques, such as rotation, zoom, and flip, are applied to both the images and masks. To ensure correct application of augmentation to both, the images and masks are stacked together as a tensor and passed through TensorFlow's augmentation layers. Subsequently, they are split apart, and the datasets are created.

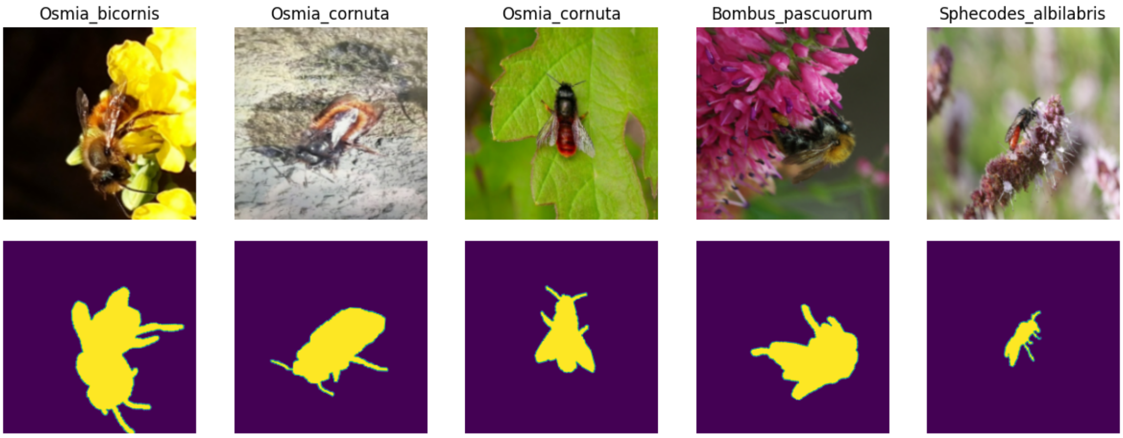

Figure 2.3. – Augmented images and masks from Bees dataset

The feature maps obtained and the masks are then normalized to have values between 0 and 1. Additionally, a z-score normalization is performed to ensure an identical mean.

During the implementation process, it became evident that similar challenges might arise during model training, particularly in the two phases of freezing and unfreezing base layers. While the classification head adapts well to the data initially, it may become unstable or settle into a local optimum when the base layers are retrained.

To mitigate this, the model is first trained with the new classification head for fixed number of epochs until it reaches an accuracy of 50-60 %, without any learning rate scheduler (i.e., without any learning rate decay). Subsequently, the entire model is trained with a lower learning rate, typically around 3e-4, using plateau decay. The specific learning rates and schedulers employed in these two phases may vary depending on the model and dataset.

3. Design of Experiments

3.1. Datasets

To validate the results of our experiments, we employed two independent datasets. The experiments were conducted separately with each dataset. Each dataset was reorganized into train, validation, and test folders, with sub folders named after the species names, and each species folder containing images corresponding to that species, following a standardized format commonly used for ease during implementation.

The first dataset,known as the Bee dataset [Chi22], comprises 25 wild bee species. This dataset was curated as part of a study investigating the 75% reduction in insect biomass observed in Germany over the past three decades. This is smaller dataset close to 800 images.

The second dataset, referred to as the Flowers dataset [NZ08], encompasses 102 species commonly found in the United Kingdom. Each class comprises between 40 and 258 images, featuring significant variations in scale, pose, and lighting. This is larger dataset with 8000 images. Both datasets were utilized in our project to simulate a classification task.

Both theses dataset are re-organised into train, val, test folder containing folders with images named by their class name. which is common a standard format for ease during the implementation. These independent datasets were crucial for validating the robustness and generalization of our experimental results. By conducting the experiments with each dataset separately, we ensured the reliability of our findings across diverse scenarios.

3.2. Metrics

The following metrics are measured at each iteration of the experiment:

- Convergence point (epochs required for convergence): important for assessing the affect of spatial loss on convergence time.
- Loss at convergence: important for assessing its ability to make accurate predictions on new, unseen examples.
- accuracy : Validation, train and test accuracy of prediction after training.

3.3. Baseline

During the baseline experiment, the model without a spatial loss term was analyzed. The multiple-input and multiple-output model, as described in the design of the experiment, was utilized, and the spatial loss term was multiplied by a factor of zero ($\beta = 0$), effectively running the model without a spatial loss term. This step was taken to ensure that the model itself remained unchanged during the experiment.

The model was trained in two phases: firstly, the new classification head was trained alone, and secondly, the entire model was trained. It was observed that the model's classification accuracy improved reasonably and the spatial loss remained almost constant with slight oscillations. The MobileNetV3Small model, which is already trained on images, produces feature maps that closely resemble the actual resized masks at layers close to the classification head. However, these feature maps did change but did not seem to improve with the training of the entire model. The spatial loss remained in the ranges as before training.

The first four images of the seven in Figure A.3 and Figure A.4 are part of the dataset containing image, masks and resized mask, while the last three are feature maps extracted at extraction points at layers with indices 61,129 and 197 respectively. It is an interesting observation to note that the feature map at deeper layers resemble the expected even before unfreezing the base model, denoting the mobilenet model

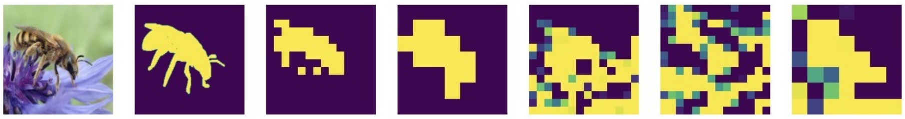

Figure 3.1. – Initial feature maps on an image from test data of Bees dataset

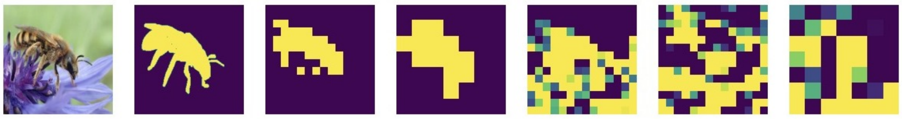

Figure 3.2. – Feature maps after trainig with zero spatial loss

is already good at capturing the features, these seem to have updated during the re-training which are different as the spatial loss term is not added.

3.4. Experiments

The two step training process mentioned in the baseline experiment is repeated, this time with the spatial loss. During the second phase, the entire model is trained with spatial loss and classification loss. As described in the fig: 2.2 feature maps are extracted at one of the extraction points. Followed by aggregation and normalization. Aggregation is done to reduce the dimension to a single channel, single channel image is compared to the resized mask to compute the spatial loss. Aggregation is performed by simply taking the average across the channels.

The loss is computed as

$$\text{Overall Loss} = \alpha \times \text{Classification loss} + \beta \times \text{Spatial loss}$$

The Overall loss is back- propagated and weights are updated to have the lower loss.

The experiment is repeated with different weighting factors (0.1, 0.5, 1, and 5) applied to the spatial loss at a single extraction point. Metrics were recorded to determine the impact of varying weighting factors on the performance of the model.

Subsequently, the best weighting factor identified from the previous experiment was utilized at different extraction points, and metrics were recorded to evaluate the consistency of the results across various extraction points.

To further validate the consistency of the findings, the experiment is replicated using the same weighting factor on the flowers dataset.

4. Results and Discussion

4.1. Feature Maps analysis

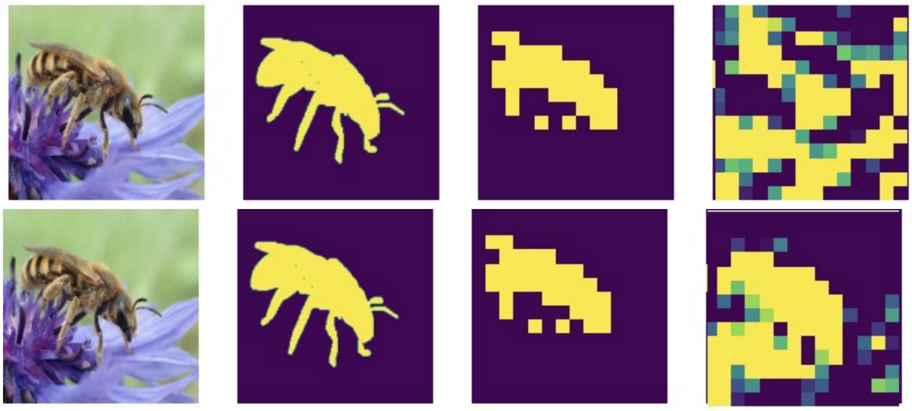

Figure 4.1. – Feature map comparison at intermediate extraction point

The fig. 4.1 shows the comparison of feature map before and after training at intermediate extraction point i.,e layer with index 129, the first and second are image and masks from test data. The other two images are resized mask and normalized feature map. It could be observed, the normalized feature map transformed and resemble similar to the expected resized feature map.

The fig. 4.2 shows the comparison of feature map before and after training, extracted at a deeper layer i.,e layer with index 197. The feature maps are transforming to resemble expected, this is an expected behaviour. Gradual reduction in spatial loss is also observed during the training .

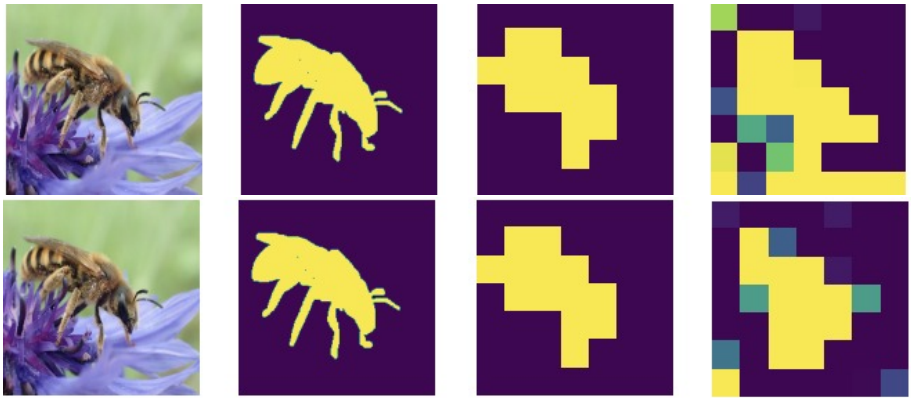

Figure 4.2. – Feature map comparison at late extraction point

4.2. Analysis of metrics

The table: 4.1 presents a comparison of the metrics with a different weighting factor to spatial loss. With weighting factor the experiment is repeated 5 times and average is considered. Standard variation is also noted to have an idea over the variation range.

It can be observed that with weight 0.1, 0.5 and 1, the tets, train and validation accuracies have slightly improved with slightly lower standard deviation with a few outliers. The Train, test, validation loss also have reduced. With weight 5 the performed worsed, it is observed the spatial loss decreased drastically and affected the classification loss, thus reducing the performance.

The table: 4.2 and table: 4.3 presents a comparison of the metrics with a 0.5 weighting factor to spatial loss and without spatial loss. It is noted that performance has slightly degraded while the convergence point was reached comparatively faster. The average number of epochs required to reach the optimum has reduced from 25 to 18 and 27 to 22.

In the table:4.4, analysis on flowers dataset can be observed to have similar results, the accuracy improved slightly and epochs required to train have reduced upon addition of spatial loss.

Table 4.1. – Comparison of Metrics with Different Weighting to Spatial Loss at early extraction point on Bees Dataset

Average of 5 measurements										
Metrics	Weight(0)		weight (0.1)		**weight (0.5)**		weight (1)		weight (5)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Test accuracy (%)	71.50	0.036	70.41	0.049	**72.32**	0.035	71.50	0.031	66.30	0.044
Validation accuracy(%)	76.43	0.169	77.80	0.038	**77.53**	0.033	79.17	0.013	67.94	0.038
Test label loss	1.18	0.04	1.18	0.15	**1.05**	0.13	1.04	0.15	1.29	0.22
Validation loss	0.94	0.17	0.91	0.11	**0.93**	0.18	0.93	0.15	1.36	0.17
Convergence	30.0	7.72	29.4	7.57	**28.4**	3.20	35.0	6.42	24.2	5.74
Train Accuracy(%)	98.8	0.007	99.0	0.006	**98.9**	0.005	99.1	0.003	98.6	0.002
Train loss	0.048	0.024	0.044	0.015	**0.046**	0.016	0.042	0.011	0.058	0.012

Table 4.2. – Comparison of Metrics with Different Spatial Loss Weights on middle aggregates on Bees Dataset

Average of 5 measurements				
Metrics	Weight(0)		weight (0.5)	
	Mean	Std	Mean	Std
Validation accuracy(%)	76.99	0.0381	73.70	0.0102
Test accuracy (%)	70.95	0.0279	64.11	0.0796
Validation loss	0.9990	0.1333	1.1280	0.1142
Test label loss	0.9357	0.1529	1.227	0.2439
Convergence	**25.2**	5.114	**18.6**	5.462
Train Accuracy(%)	98.89	0.0023	98.55	0.0023
Train loss	0.0494	0.0071	0.0698	0.0095

Table 4.3. – Comparison of Metrics with Different Spatial Loss Weights on late aggregates on Bees Dataset

Average of 5 measurements				
Metrics	Weight(0)		weight (0.5)	
	Mean	Std	Mean	Std
Validation accuracy(%)	76.16	0.0438	75.61	0.0339
Test accuracy(%)	70.41	0.0509	70.13	0.0515
Validation loss	1.0250	0.1871	1.0029	0.1517
Test label loss	0.9667	0.1296	0.9397	0.1939
Convergence	**27.4**	7.9145	**22.2**	4.1665
Train Accuracy (%)	98.89	0.0060	98.72	0.0051
Train loss	0.0494	0.0216	0.0608	0.0184

Table 4.4. – Comparison of Metrics on flowers dataset at late extraction point on Flowers Dataset

Metrics	Weight(0)	weight (1)
Validation accuracy(%)	93.76	94.00
Test accuracy(%)	93.76	94.80
Validation loss	0.3085	0.3110
Test label loss	0.2861	0.2877
Convergence	**50**	**40**
Train Accuracy (%)	100	100
Train loss	7e-4	5e-4

4.3. Analysis of Loss history

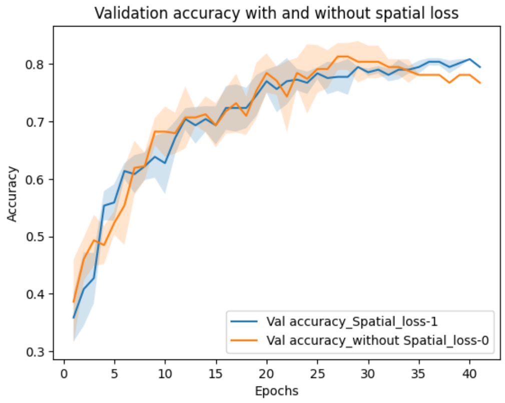

Figure 4.3. – Validation accuracy- Bees dataset at Early extraction point

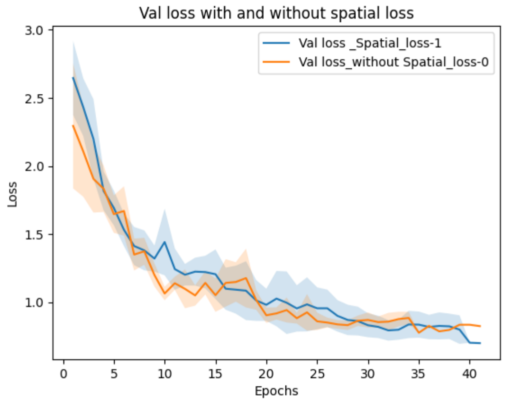

Figure 4.4. – Classification validation loss- Bees dataset at Early extraction point

The fig:A.1 and fig:A.2 is the average of histories with standard deviation shading on Bees dataset, with feature maps from early extraction point. The minimum of length of the epochs in the training history is considered and all the histories are trimmed to the minimum. At each epoch, the average across the iterations and standard deviation is calculated. Average is plotted and shading with width as standard deviation is provided on both sides. The fig:A.1 compares the Validation accuracy with and without spatial loss during the training. The fig:A.2 compares the Validation loss with and without spatial loss during the training.

4.4. Analysis of training log

Analysing the Training log at the first and last epoch, it could be observed that, loss is sum of output label loss and spatial loss (at late extraction point). The overall

```
Epoch 1/50
73/73 - 33s - loss: 2.0103 - output_label_loss: 1.7869 -
 late_aggregate_loss: 0.2235 - output_label_accuracy: 0.4914 - val_loss
 : 3.0653 - val_output_label_loss: 2.8943 - val_late_aggregate_loss:
 0.1710 - val_output_label_accuracy: 0.2329 - lr: 3.0000e-04 - 33s/
 epoch - 456ms/step

Epoch 29/50
73/73 - 18s - loss: 0.1126 - output_label_loss: 0.0357 -
 late_aggregate_loss: 0.0769 - output_label_accuracy: 0.9948 - val_loss
 : 1.3875 - val_output_label_loss: 1.3047 - val_late_aggregate_loss:
 0.0828 - val_output_label_accuracy: 0.7260 - lr: 1.2000e-05 - 18s/
 epoch - 241ms/step
```

loss decreased and accuracy increased during the training. Initially the Classification loss (output label loss) is considerably larger value compared to the spatial loss. At the end of the training the classification is smaller then spatial loss. Both loss values have reduced considerably during the training. It is important to note that spatial loss contributes more to the loss compare to the classification loss. As the classification has reduced considerably.

5. Summary and Outlook

In summary, Classification model train based on the classification loss, which is computed by comparing predicted probabilities and ground truth labels, in this an study spatial loss is added, to analyse any potential positive impacts. From the results of the extensive experiments, it can be said that spatial loss did have an impact on the training process and outcome. Addition of a small portion of spatial loss to over loss had positive impact on accuracy and convergence. The same has been verified by using the feature maps at multiple extraction points. In all cases, the model converged faster.

5.1. Future Avenues

Dynamic or Adaptive weighting: As noted in the earlier section 4.4, As the training progresses the initially high value of classification reduces at a faster rate and spatial loss reduces at small rate which leads to spatial loss having more impact compared to classification loss. Which causes Classification loss and accuracy to platue as the loss is majorly spatial loss. It can be observed that spatial which contributes roughly to 10% of the overall loss at the starting of training is 2/3 at the last epoch.

It would be more beneficial to have a dynamic weighting factor, which resales the value of spatial loss to have less contribution to overall spatial loss. A callback could be implemented to achieve this.

Alternative Architectures: The MobileNetV3Small architecture is an established model specifically designed for mobile and edge devices. As illustrated in Figure

2.1, it is characterized by its deeper structure and increased dimensions across the channels, resulting in smaller feature map sizes.The key idea of the study is to use feature maps extracted from hidden layers of classification model to guide the model. The significance of this approach may be more pronounced in models with higher-dimensional feature maps or in models built from scratch.

Mask Dilation: Dilation is an operation applied to masks to expand their size, particularly useful when estimating object positions rather than their exact coordinates. Dilation of masks before normalization and resize may yield intriguing results, with the selection of dilation parameters and patterns potentially influencing outcomes differently.

Alternate Spatial loss: While MSE loss is primarily considered for analysis, alternative loss functions such as IoU may be more suitable for this model's characteristics.

6. Conclusion

The study aimed to analyze the effect of spatial loss on classification models and produced some interesting results. The study confirmed that the addition of spatial loss did have an effect on the classification loss. By comparing models trained with weighting factors of spatial loss terms, it has been observed that a smaller proportion of addition of spatial loss resulted in a acceptable increment in accuracy and a significant improvement in the number of epochs required to converge. Conversely, larger proportions of spatial loss had the opposite effect. This can be reasoned as the classification loss is small, which produces small gradients to tune towards optimal loss.

This indicates that spatial loss functions could aid in guiding the classification models towards better and faster convergence.

A. Appendix

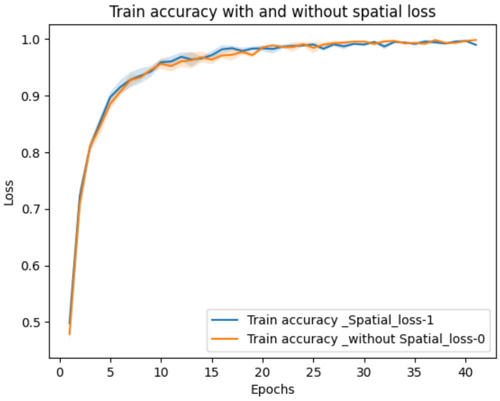

Figure A.1. – Training accuracy-Bees dataset at Early extraction point

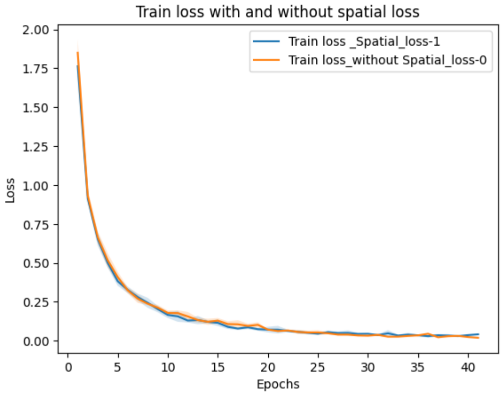

Figure A.2. – Classification Training loss- Bees dataset at Early extraction point

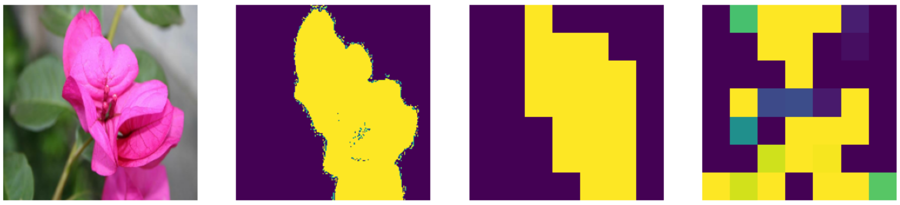

Figure A.3. – Initial feature maps on an image from test data of Flowers dataset

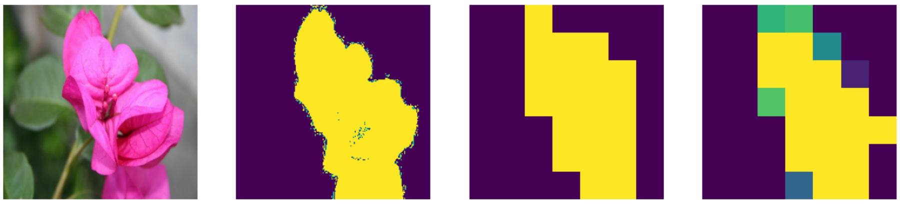

Figure A.4. – Feature maps after training with spatial loss (with weight 1)

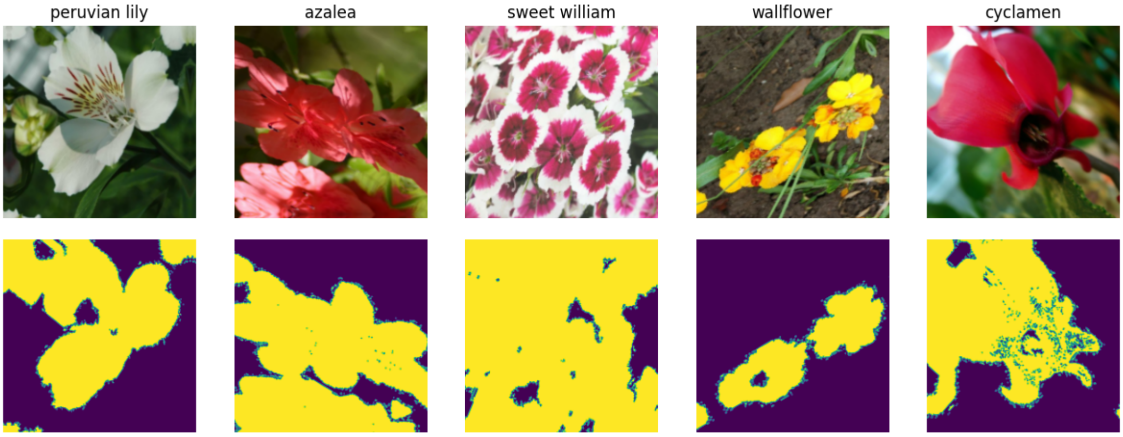

Figure A.5. – Samples of augmented images and masks from Flowers dataset

List of Tables

List of Figures

Bibliography

[AKB22] Amrani, Elad ; Karlinsky, Leonid ; Bronstein, Alex: *Self-Supervised Classification Network*. 2022 1

[Chi22] Chiaburu, Teodor: *beexplainable: XAI Experiments on an Annotated Dataset of Wild Bee Images*. https://github.com/teodorchiaburu/beexplainable, 2022 3.1

[DMH17] Dwibedi, Debidatta ; Misra, Ishan ; Hebert, Martial: Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In: *CoRR* abs/1708.01642 (2017). http://arxiv.org/abs/1708.01642

[GBC16] Goodfellow, Ian ; Bengio, Yoshua ; Courville, Aaron: *Deep Learning*. MIT Press, 2016. – http://www.deeplearningbook.org 1

[LGG+17] Lin, Tsung-Yi ; Goyal, Priya ; Girshick, Ross ; He, Kaiming ; Dollár, Piotr: Focal Loss for Dense Object Detection. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, S. 2999–3007 1

[NZ08] Nilsback, M-E. ; Zisserman, A.: *Flowers-102: 102 Category Flower Dataset*. https://www.robots.ox.ac.uk/~vgg/data/flowers/102/, 2008 3.1

[RBPGY+21] Rey-Barroso, Laura ; Peña-Gutiérrez, Sara ; Yáñez, Carlos ; Burgos-Fernández, Francisco J. ; Vilaseca, Meritxell ; Royo, Santiago: Optical Technologies for the Improvement of Skin Cancer Diagnosis: A Review. In: *Sensors* 21 (2021), Nr. 1. http://dx.doi.

org/10.3390/s21010252. – DOI 10.3390/s21010252. – ISSN 1424–8220 1.1, 1, A

[ZF13] ZEILER, Matthew D. ; FERGUS, Rob: *Visualizing and Understanding Convolutional Networks.* 2013 1

Declaration Of Authorship

I, Uday Kaipa, hereby affirm, that I have prepared the present work without the unauthorized help of third parties and without the use of any aids other than those specified. The data and concepts taken directly or indirectly from sources are marked with a reference to the source. This thesis has not been submitted to an examination board in the same or a similar form, or published in any other way, either at home or abroad.

Ilmenau, 2024/03/01

Uday

Uday Kaipa