Eberhard Karls University Tübingen
Faculty of Science

Gemstone Classifier

Report
Bildverarbeitung / Machine Learning

Authors:
Stephan Amann (5445826),
Tanja Huber (5471421),
David Kleindiek (5479331)

March 15, 2024

Abstract

To simplify the task of gemstone classification a simple and lightweight CNN has been modeled. Training data was scraped from an online store to be able to cover a wide variety of gemstones. Gemstone types that were further differentiated by the region they were sourced from and gemstones that show a variation of characteristics in one class were found to be the most challenging to classify. Although top-1 accuracy varied, top-3 and top-5 accuracies remained consistently high across approximately 100 classes for fine-tuned pre-trained state of the art models as well as the newly created one. One suggested improvement involves dividing the model into two networks: one for general classification and another for regional differentiation.

Contents

1 Introduction

Depending on the considered source, gemstone classes are estimated anywhere between a hundred and several hundred. Many gemstones share similar features such as color, internal structures or refractive indices but can nonetheless be classified differently. These gemstone classes can oftentimes be further separated by the region they were found in. (International Gem Society, 2024a; Jeweller Pawnbroker, 2024)

As such it can be a difficult task even for professionals to correctly identify a gemstone, let alone for a layman. Even if several metrics of a gemstone are known there may still be several possible options a gemstone could be classified as. Common metrics that are comparatively simple to collect include color, transparency, refractive index and density. However, since these metrics can be insufficient to surely identify a gemstone, additional testing may be required. When refractive index readings are difficult to judge this can include spectroscopic analysis to study the absorption and emission of light. Bifringence, in which case several refractive indices apply to one gemstone, or plechorism, which causes the angle a gemstone is looked at form to change the perceived color of a gemstone, can be be investigated. Ultraviolet light might be used to study specific interactions as well. (International Gem Society, 2024b; The Gemmological Association of Great Britain, 2022)
In addition to the category of a gemstone, it can furthermore be difficult to distinguish whether or not a gemstone was naturally or synthetically created. This typically involves observing inclusions. (Ceylons Munich, 2023; The Gemmological Association of Great Britain, 2022)

Many of these tests for gemstone classification include tools that may very well not be accessible to a layman. Even for professionals identifying and then possibly also matching gemstones to each other to create a piece of jewelry can be a time consuming endeavour, as it might also include the consideration of the angle a gemstone is placed at (Geodes, 2024).

As such a tool to simplify the classification of gemstones could help not only laymen but even professionals. Neural networks are more and more commonly used as helpful tools for various tasks. This includes classification ranging from pastries over animals and flowers to tumors (Consumer Technology Association, 2021; Technische Universität Ilmenau, 2024). The plant identification app Flora Incognita from Technische Universität Ilmenau (2024) specifically motivated this project, as it is easily accessible and creates a quick and simple learning opportunity by including various information about the identified plants. Following from that, and an interest in gemstone classification, this project focuses on creating a small and lightweight classification model, which could ideally be implemented as an app to make gemstone classification approachable to anyone.

Figure 1.1: Top row: same cut, different gemstones. Bottom row: different cuts, same gemstone.

Since it is realistically more likely to come across already cut gemstones, only those will be considered for this project.
Other than the density and hardness of a gemstone limiting cuts to the gemstone, there are no combinations that are unreasonable. As such the cut of a gemstone should not affect the learning of the network. In Figure 1.1 an arrangement of different gemstones with the same cut and same gemstones with different cuts is shown.

To judge the efficiency of the created model a comparison to deeper and more complex state of the art models is conducted.

2 Fundamentals and Related Work

2.1 Canny Edge Detector

The canny edge detector is used to detect edges and create output images displaying the found edges. First of all, a Gaussian filter is applied to smooth the image and remove noise. A 2D first derivative operator is applied to find the intensity gradients of the image. Tracking along high intensity ridges allows for marking only the height of the ridges and suppression of surrounding noise. This is done using a double threshold, with the higher threshold deciding when tracking is started and the lower one deciding when it is stopped. Finally, all edges deemed weak or with no connections to strong edges are dropped. (Wikipedia, 2023)

2.2 Artificial Neural Networks

Artificial neural networks (ANNs) are based on biological neurons. They are created through receiving some input into a weighted, interconnected network of neurons, activating based on specific activation functions. Each neuron sends output signals to neurons further down the network based on its received input signal and the activation function describing its internal state. Loss functions are used to quantify the difference between the predicted output and the ground truth. During the training of a network the parameters are adapted to minimize this loss. (Wikipedia, 2024b)

2.3 Convolutional Neural Networks

A Convolutional neural network (CNN) focuses on inputs with grid structures, which are most commonly images. Thus specific techniques are used to compute the connections between the grid layers.
Convolutional layers compute a dot product between two matrices, namely a filter kernel and a portion of the input matrix. The filter kernel consists of parameters which the network learns. Stride decides the step-size with which the filter kernel is moved across the matrix.
Pooling layers summarize nearby pixels together, reducing the size of their input. Most common are maximising or average pooling layers.
Fully connected layers are used to ensure that all neurons are considered for evaluation together. These layers have connections between all neurons of their input and output and are commonly used in the last layers of a network. (Wikipedia, 2024a)

2.4 Regularization techniques

2.4.1 Dropout

Dropout is a regularization technique for neural networks that reduces overfitting by randomly setting a specified proportion of neuron activations to zero during each training iteration. This forces the network to learn redundant representations, making it less reliant on specific neurons and patterns. (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014)

2.4.2 L2 Weight Decay

L2 regularization also known as ridge regression combats overfitting by adding a penalty term to the loss function that is proportional to the sum of the squared weights in the network. This encourages smaller weights through penalizing large weight values and prevents one weight from dominating the model. (Cortes, Mohri, & Rostamizadeh, 2009)

2.5 Evaluation Metrics

In the following, various possible evaluation metrics for neural networks and their use cases are summarized.

2.5.1 Confusion Matrix

Confusion matrices are used to give an overview over the classification performance of a model. This is done by displaying all predicted classes against on one axis of a matrix and all true classes on the other axis of a matrix. Values in the cells denote the amount of times a specific case occurred. This can be done for binary classification as well as multi-class classification.

2.5.2 Accuracy

Accuracy measures the overall percentage of times the classifiers prediction was correct. It's a useful metric when you have a balanced dataset - approximately equal numbers of images for each class - and the goal is simply to maximize overall correct classifications. (Cloud Factory, 2024)

It is calculated by the following formula:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

2.5.3 Precision

Precision is a metric for evaluating the performance of a classification model, especially when the cost of false positives is high. It is particularly useful for class-wise evaluation, as it allows for the assessment of a model's ability to correctly identify instances of a specific class without being misled by the presence of instances from other classes. (Evidently AI, 2024)

Precision is calculated by the following formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

A precision of one would thus mean that the target class is always predicted correctly.

2.5.4 Recall

Recall shows if a model can find all instances of a target class and is an important metric if the cost of false negatives is high. It is most commonly used in detection tasks to judge whether or not a target has been found. (Evidently AI, 2024)

It is calculated by the following formula:

$$\text{Recall} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN}}$$

Is the recall equal to one, all instances of the target class were found.

2.5.5 Intersection over Union

Intersection over union (IoU) is used for image segmentation tasks and investigates how well a predicted bounding box matches a ground truth bounding box. (Adrian Rosebrock, 2022)

It is calculated with the following formula:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

2.6 Pre-Trained Models

This section explores several state-of-the-art CNN architectures suitable for image classification tasks. These pre-trained models offer a powerful foundation for transfer learning, leveraging their knowledge gained on the vast ImageNet dataset to accelerate training and improve performance on new classification problems.

2.6.1 VGG16

VGG16 was created in 2015 at the University of Oxford and won the ImageNet Challenge in 2014. The basic idea was testing and proving whether a deep network with small filter kernels could outperform more shallow networks with larger kernels. The total model has 19 layers, with 16 of them being convolutional layers and the last three being fully-connected layers. (Simonyan & Zisserman, 2014)

The specific architecture uses very small 3x3 convolution filter kernels to increase the depth. The convolution layers are followed by 2x2 max pooling layers each. ReLU activation functions are used. Finally, 3 fully-connected layers and a softmax finalize the model. (Simonyan & Zisserman, 2014)

Having more but smaller convolution filter kernels lead to more non-linear rectification layers which made the decision function more discriminative. The smaller kernels furthermore function like a regularization compared to bigger convolution filters, resulting in less overall parameters. (Simonyan & Zisserman, 2014)

VGG16 uses a L2 weight decay of $5 \cdot 10^{-4}$ and a dropout of 0.5. (Simonyan & Zisserman, 2014)

2.6.2 ResNet50

ResNet was created in 2015 and won the ILSVRC classification task in the same year. It introduced residual functions, which are also known as skip connections. (He, Zhang, Ren, & Sun, 2016)

The residual functions connect input layers to layers deeper in the model, ensuring that the model learns the difference between input and output. They are applicable whether the inputs have the same size or not, using zero padding or projections in case the sizes differ. (He et al., 2016)

The specific architecture uses mostly 3x3 convolutional filters and achieves downsampling through stride 2 instead of max pooling layers. The last layers of the model consist of a global average pooling layer followed by a fully connected layer and a softmax function. (He et al., 2016)

The convolutional layers follow two specific design rules: If input and output have the same feature map size, the layers have the same amount of filters. If the size of the output is halved, the amount of filters is doubled. By doing so the time complexity per layer is kept consistent. (He et al., 2016)

ResNet uses a weight decay of 10^{-4}. Overall the model is deeper but less complex compared to VGG (He et al., 2016).

2.6.3 MobileNet

The original MobileNet was presented in 2017 with the goal of creating a small and lightweight model which is able to achieve a trade-off between accuracy and latency (Howard et al., 2017).

MobileNet is based on depthwise separable filter kernels, which split the standard convolution layer into two layers. The first layer filters the input, using depthwise convolution which applies a single filter kernel to each input channel. The second layer combines its input through a 1x1 convolution. This combination of layers reduces model size as well as computational costs. The very first layer in the model is a full convolution layer. Downsampling is implemented through stride in the first layer and the following depthwise convolutions. Batch normalization and ReLU activation functions are used. Average pooling is done before the final fully connected layer. If the depthwise separable filters are counted separately the model results in 28 layers total. (Howard et al., 2017)

MobileNet further introduces a width multiplier to thin the model at each layer by multiplication, as well as a resolution multiplier which can be applied to the input image to set the resolution. (Howard et al., 2017)

Very little to no weight decay is required since there are few parameters used for the depthwise filters (Howard et al., 2017).

3 Methods

3.1 Data Aquisition

Searching for available datasets, no sufficient dataset could be found. The two free datasets found only covered a comparatively small subset of gemstones and offered limited images per class. With the dataset on Kaggle covering 87 classes with 3 200 images total and another dataset only covering 20 classes (Daria Chemkaeva, 2020; Päivi Lujala, 2009). Since generalization is required to be able to ensure good classification of new images, more images per class were deemed necessary for this project. As such, own data was collected.

While wiktionaries and informative websites covered all classes of gemstones, only few images were provided per class. Furthermore, the images also included uncut gemstones, which were unwanted for this project. Even combining several wiktionaries did not result in a sufficient amount of images.

The best coverage of gemstones and images, specifically of the most wanted and common gemstones, was found in online stores. Taken into consideration were Edelsteine.at (https://www.edelsteine.at/), Gemselect.com (https://www.gemselect.com/) and Gempundit.com (https://www.gempundit.com/).

Edelsteine.at was not chosen since it was limited to 59 classes of gemstones. While Gemselect.com showed good coverage of gemstone classes, the images provided differed in background and oftentimes including several gemstones in one picture. In comparison, Gempundit.com had good coverage of classes with around 60 000 products resulting in 170 different gemstone classes, including differentiation by regions. Almost all images had equal and uniform backgrounds. Additonally, gemstones were displayed from different angles, presumably enabling good generalization. Thus Gempundit.com was chosen to gather data.

3.2 Data Scraping

A custom parser to collect all images was written, using Python and the BeautifulSoup4 library. For all available gemstone classes image download links were gathered by navigating through the structure of the website and then downloaded afterwards in a separate step. The code of this can be accessed through https://github.com/SATHDKTT/bv-ml-project.

Gathering the data lead to many unexpected challenges, mostly due to the accessibility of the website. One download request required about 3 seconds to load. Parallelization

of requests resulted in exceeding the website's request rate limit and thus temporary IP blocks. Multi-threading with random delays turned out to be the best solution to gather all images.

In total over 140 000 images were downloaded, resulting in 8.5 GB. Later on it was observed that the supply of the store is changing not only in regard of images per product, but also products and thus gemstone categories themselves. However, this only concerned comparatively few classes which would likely not be as common to be traded and thus occur less frequently overall. As such, the first collected data was used and not updated throughout the project. Nonetheless, it should be noted that if the project is reproduced, this would likely cause some differences in regard to the results.

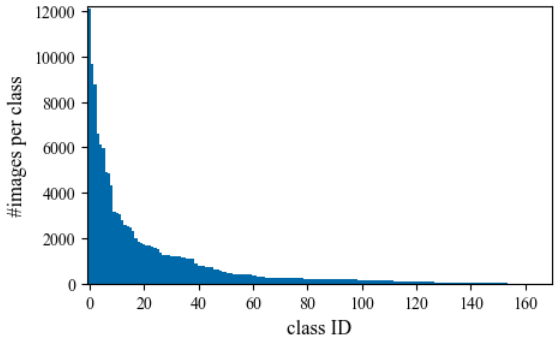


Figure 3.1: Distribution of gemstone classes after data cleaning.

Depicted in Figure 3.1 is the distribution of images per gemstone class. The 170 obtained classes vary from over ten-thousand images for popular stones such as "Sapphire" or "Emerald" to below a hundred images for less popular stones like "Malachite".

3.3 Image Preparation

The collected data was cleaned up and prepared to be used as input for the CNN using Python and OpenCV.

3.3.1 Data Cleaning

Although most images were of gemstones and with uniform backgrounds, a small subset of images contained pictures of certificates or hands. Most of these images could be removed based on their filenames. However, even then a subset of unwanted images remained. It was attempted to remove certificates using optical character recognition tools. However, this turned out to not be reliable. Similarly, detection of skin tone in large areas of the image failed to successfully identify images including hands. In the end, along with double checking the images removed based on filenames, it was manually ensured that there were no remaining images of certificates or hands in the gathered data.

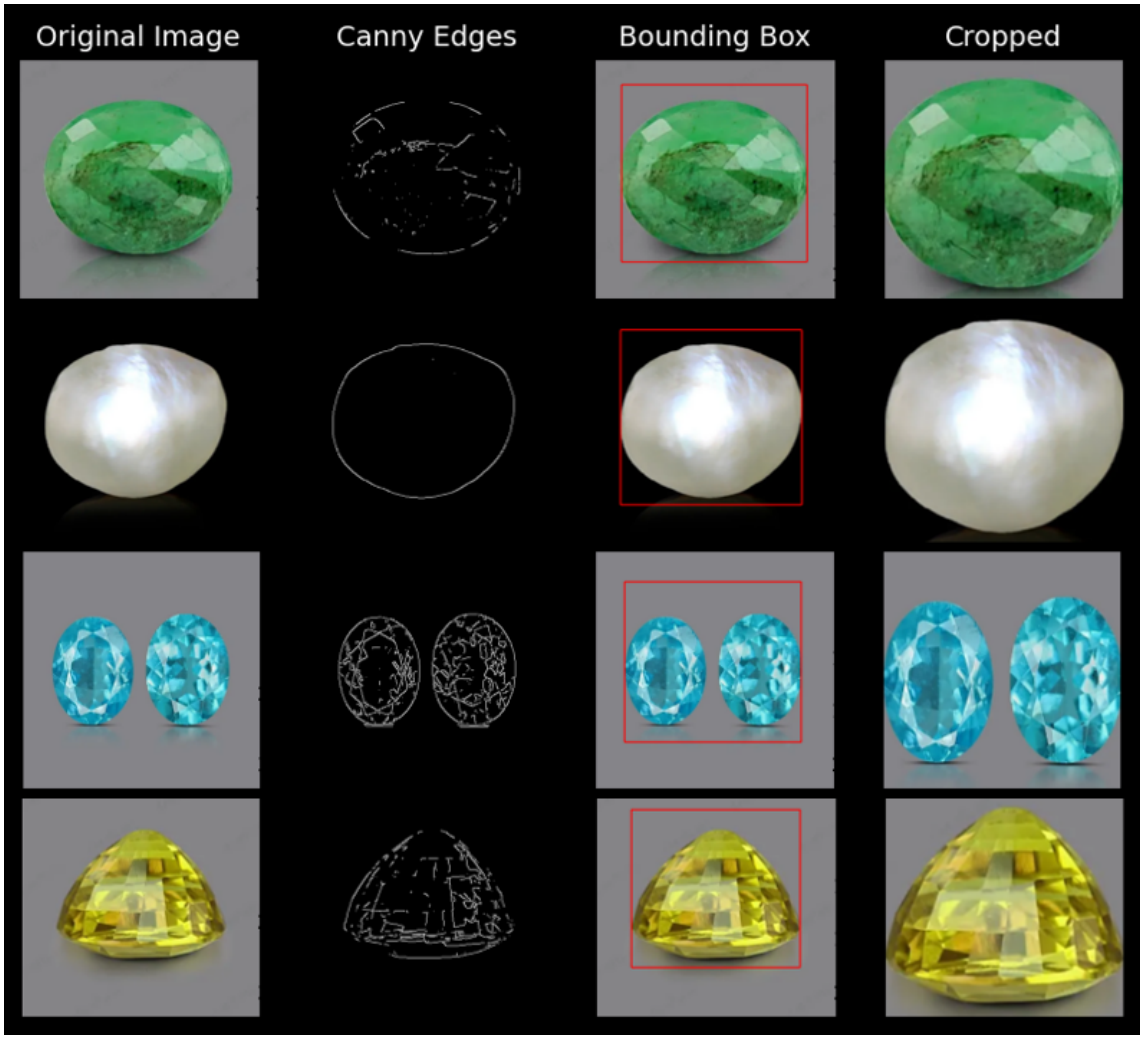


Figure 3.2: Examples of cropping from original images to RoI.

3.3.2 Cropping to Region of Interest

As visible in Figure 3.2 a large portion in each gemstone image only depicted the uniform background without any structure, including shadows. As such, the images were cropped so that only the region of interest would be handed over to the CNN. Since the backgrounds were uniform, the Canny edge detector could be used to locate the gemstone. A rectangle surrounding the gemstone is then extracted. For the CNN input, the longer side of the rectangle is scaled to 244 pixels and the shorter side enlarged accordingly to generate images of 244x244 pixels.

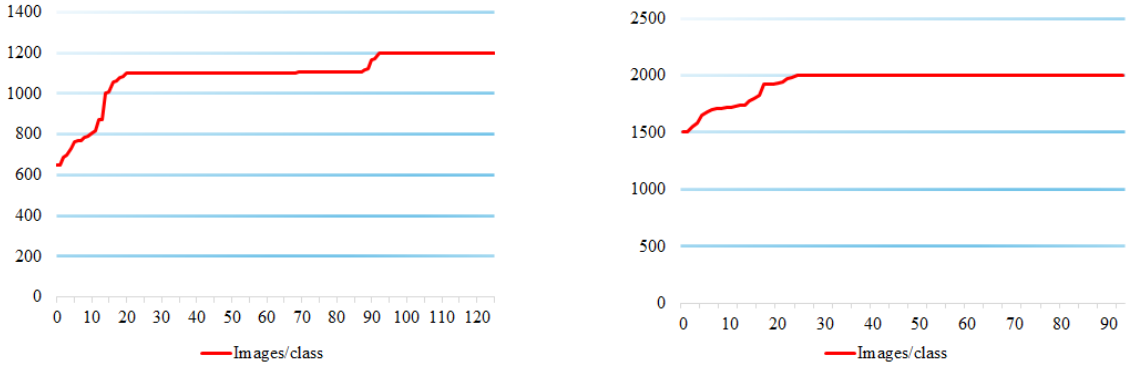
3.3.3 Data Augmentation

Data augmentation is used not only to improve the generalization of the CNN but also balance out the distribution of images per gemstone class (Figure 3.1). Since the coloring and internal structures of gemstones are crucial for identification, common augmentations such as color filtering or warping the images are refrained from to not alter relevant

information. The only augmentations used are rotation, in 45 degree steps, and mirroring. This allowed for eight times the original images, if required.

3.4 Datasets

The augmentation is specifically used to create two balanced datasets, which will later on be used to judge improvements from the network based on the amount of images it was given per class.



(a) Dataset D1k with around one thousand images per class.

(b) Dataset D2k with around two thousand images per class.

Figure 3.3: Comparison of datasets D1k and D2k.

The first dataset, referred to as D1k, aimed for one thousand images per gemstone class, but included classes up to 1 200 and down to almost 600 images. The specific distribution can be seen in Figure 3.3a.

The second dataset, further referred to as D2k, aimed for around two thousand images per gemstone class. Classes with down to 1 500 images were included. The distribution is depicted in Figure 3.3b.

For training and testing the model a validation split of 20% is used on the created dataset.

3.5 Model Construction

Tensorflow has been used to construct the new model. It follows the common input size of 224x224. It consists of five convolution layers to approximate a smaller, more lightweight architecture of comparable models, such as VGG16. For consideration of intricate details and lowering the overall amount of parameters, 3x3 kernels are used for the convolution filters. Each convolution layer is followed by a pooling operation. To extract the most prominent features, these are max pooling operations, with an exception of the fourth convolution layer having an average pooling operation to smooth the feature map and thus

improve generalization. Finally, the feature map is flattened and two fully connected layers and a softmax are used as classifier.

After fine-tuning, regularization was deemed necessary and a dropout of 0.5 and L2 weight decay were implemented. The final model can be seen in Figure 3.4.

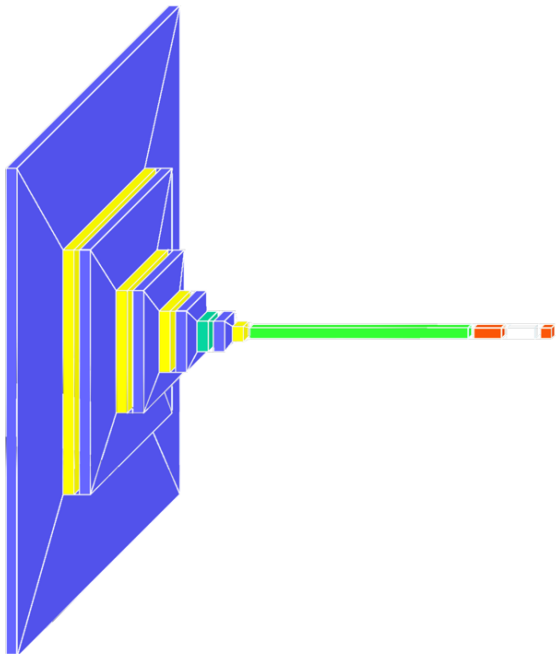


Figure 3.4: Final architecture of the created model.
Blue: Convolution layer, Yellow: Max Pooling layer, Dark green: Average Pooling layer, Light green: Flatten, Orange: Dense, White: Dropout

For faster training and lower memory requirements, a batch size of 32 was chosen. Following testing, a fast convergence could be noted, as such the model, as well as the pre-trained models, are trained for 15 epochs with a learning rate of 10^{-3} and then fine-tuned with a lower learning rate of 10^{-5} for an additional 5 epochs.

3.6 Pre-trained Models

A selection of pre-trained models consisting of VGG16, ResNet50 and MobileNet was chosen to compare the new model to:

- VGG16 has been chosen to compare a model with similar architecture but deeper layers.
- ResNet50 has been chosen to compare the usage of skip connections, as well as deeper layers.
- MobileNet has been chosen to compare the new model to another small and lightweight model.

Since it should accelerate training and only slightly lower the accuracy, transfer learning has been used. As such for the first 15 epochs the weights in the convolution layers were frozen and only the classifier was trained. For the final five epochs with the lower learning rate, all parameters were fine-tuned.

Tensorflow has also been used for accessing the pre-trained models.

4 Results

4.1 Accuracy

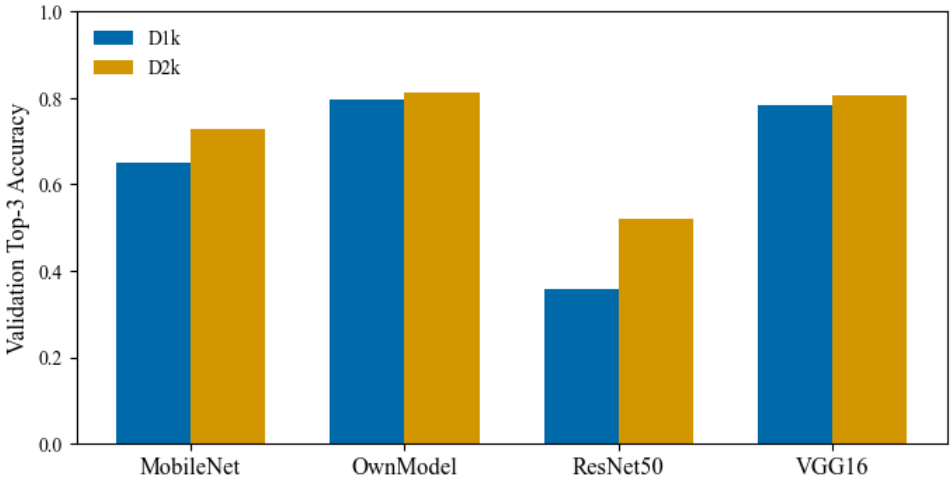


Figure 4.1: Comparison of D1k and D2k after 15 epochs with a learning rate of 10^{-3}.

The different performances of the datasets D1k and D2k are shown in Figure 4.1. This is done using the top-3 accuracy of all tested models after 15 epochs.

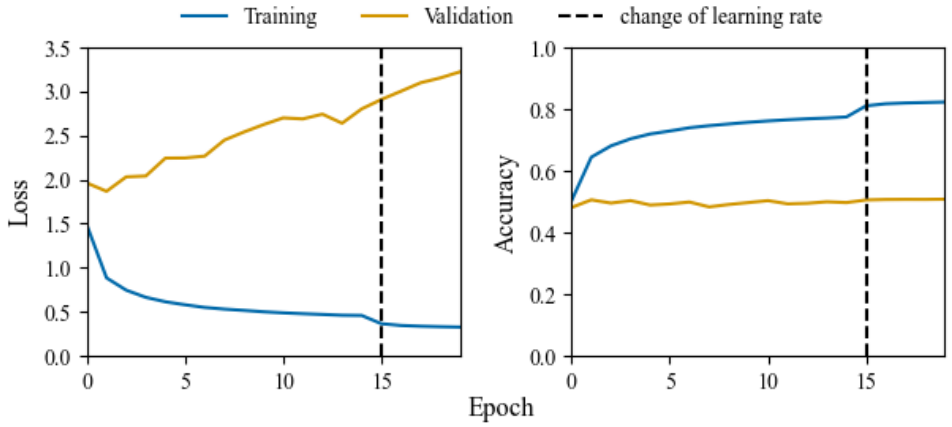


Figure 4.2: Loss and accuracy in training and validation for the first version of the created model. Learning rate is changed from 10^{-3} to 10^{-5}.

The Figures 4.2 and 4.3 show the training and validation loss over all epochs for the first version of the created model and the final regularized version of the model. For these

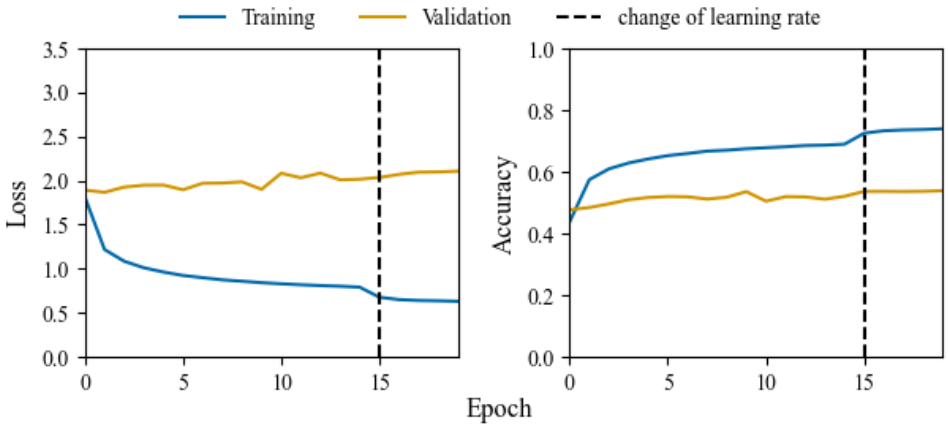


Figure 4.3: Loss and accuracy in training and validation for the final version of the created model with regularization. Learning rate is changed from 10^{-3} to 10^{-5}.

figures and all the following ones, the dashed line indicates the learning rate transition from 10^{-3} to 10^{-5}.

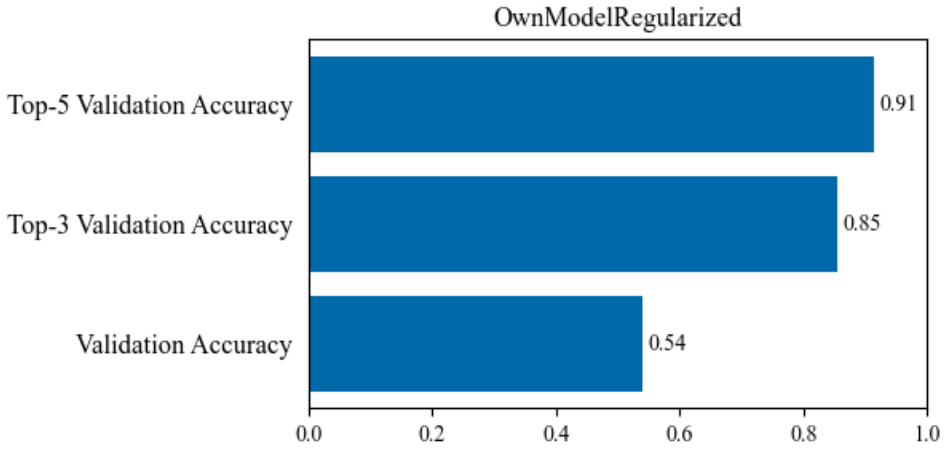


Figure 4.4: Validation accuracy, top-3 validation accuracy and top-5 validation accuracy for the own model with regularization after 20 epochs.

Shown in Figure 4.4 are the validation accuracy, top-3 and top-5 validation accuracy of the created model after the final epoch. The top-3 and top-5 validation accuracy describe whether the top-3 or top-5 predictions of the model included the ground truth gemstone class.

A full overview is given in Figure 4.5 through an arrangement of loss, accuracy as well as top-3 and top-5 accuracy. This includes all of the models training and validation per epoch.

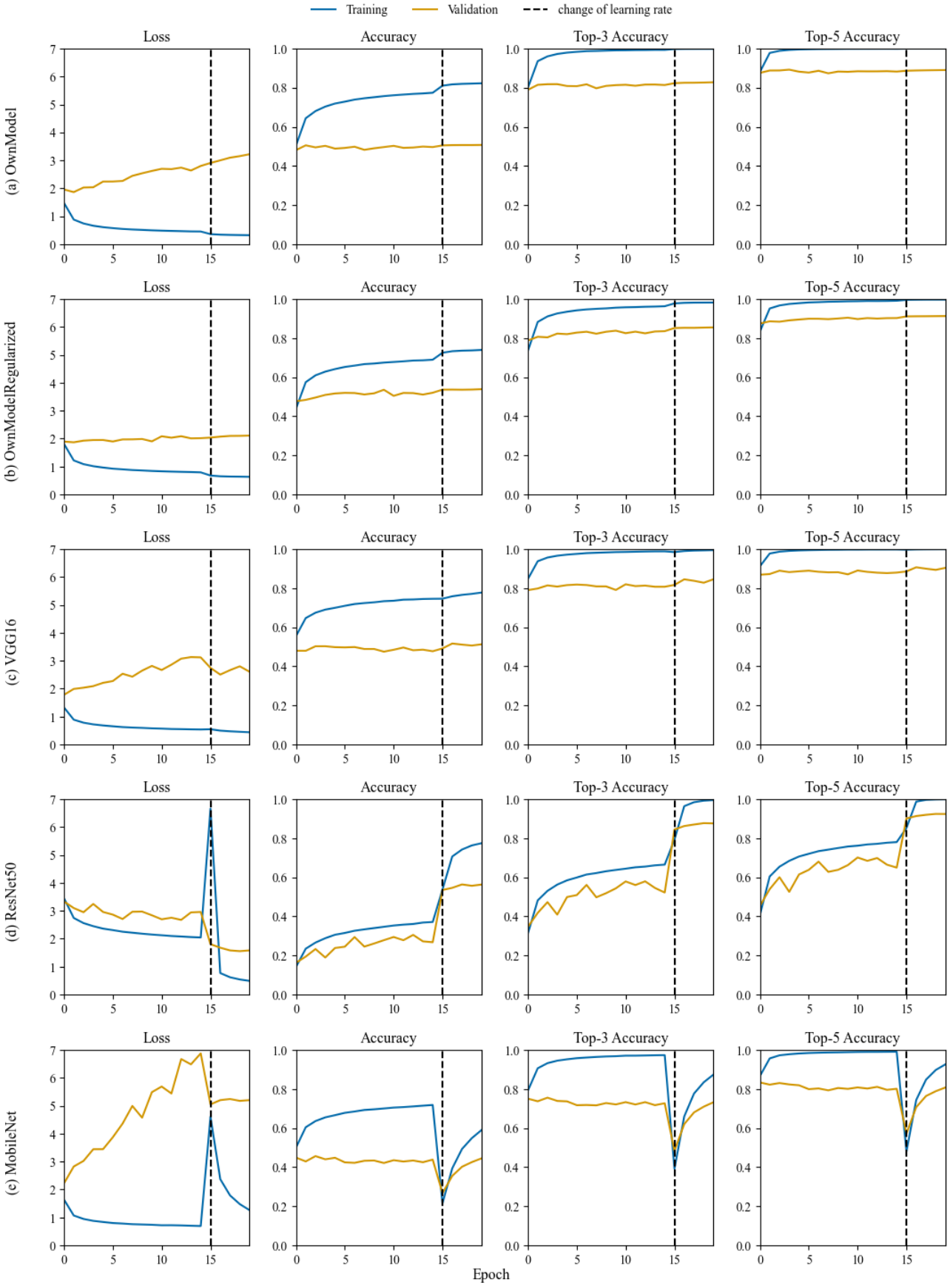


Figure 4.5: Loss, accuracy, top-3 and top-5 accuracy in training and validation for: (a) own model, (b) own model regularized, (c) VGG16, (d) ResNet50, (e) MobileNet. Learning rate is changed from 10^{-3} to 10^{-5}.

An overview over the accuracy, top-3 accuracy and top-5 accuracy per model is shown after 15 epochs with a learning rate of 10^{-3} in Figure 4.6. The same accuracies after the fine-tuning epochs with a learning rate of 10^{-5} are displayed in Figure 4.7.

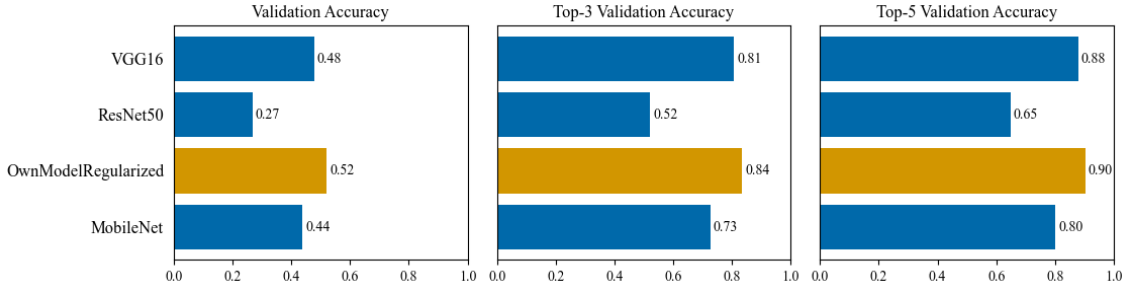


Figure 4.6: Comparison of the validation accuracy, top-3 validation accuracy and top-5 validation accuracy for all models after 15 epochs.

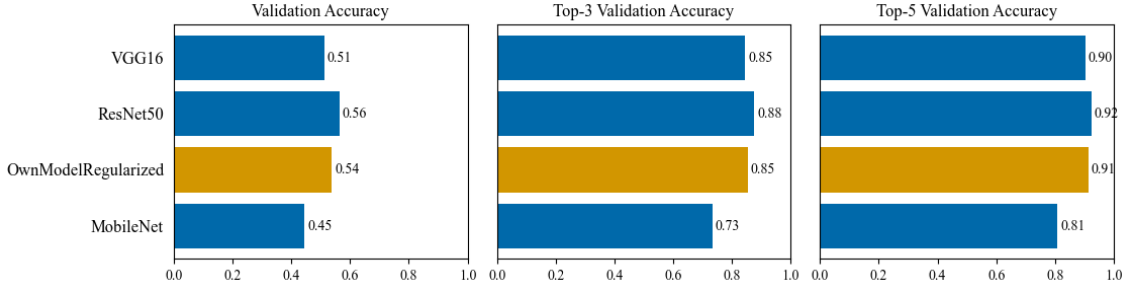


Figure 4.7: Comparison of the validation accuracy, top-3 validation accuracy and top-5 validation accuracy for all models after 20 epochs.

4.2 Precision

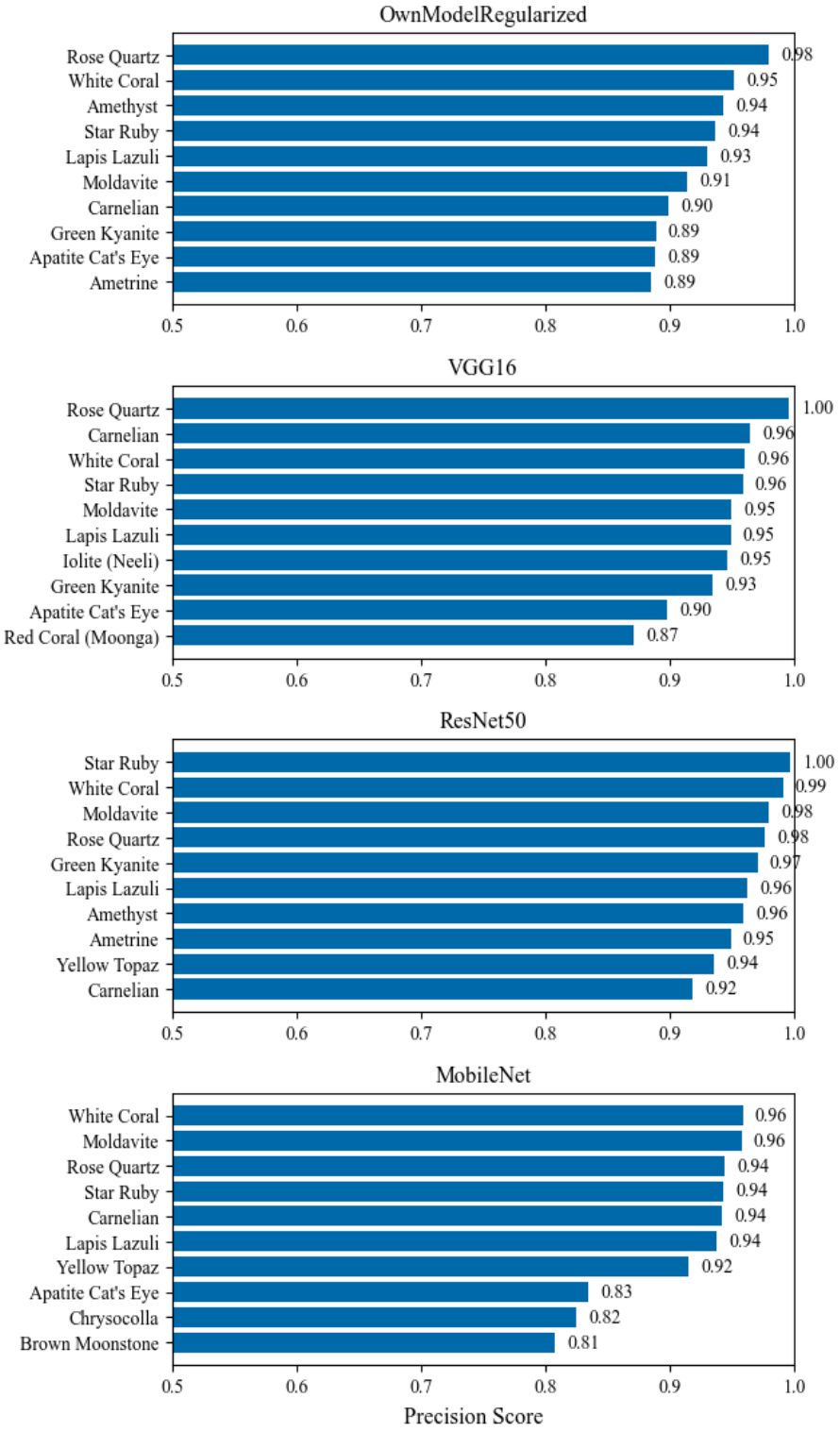


Figure 4.8: Top-10 classes with highest precision per model for: own model, own model regularized, VGG16, ResNet50, MobileNet.

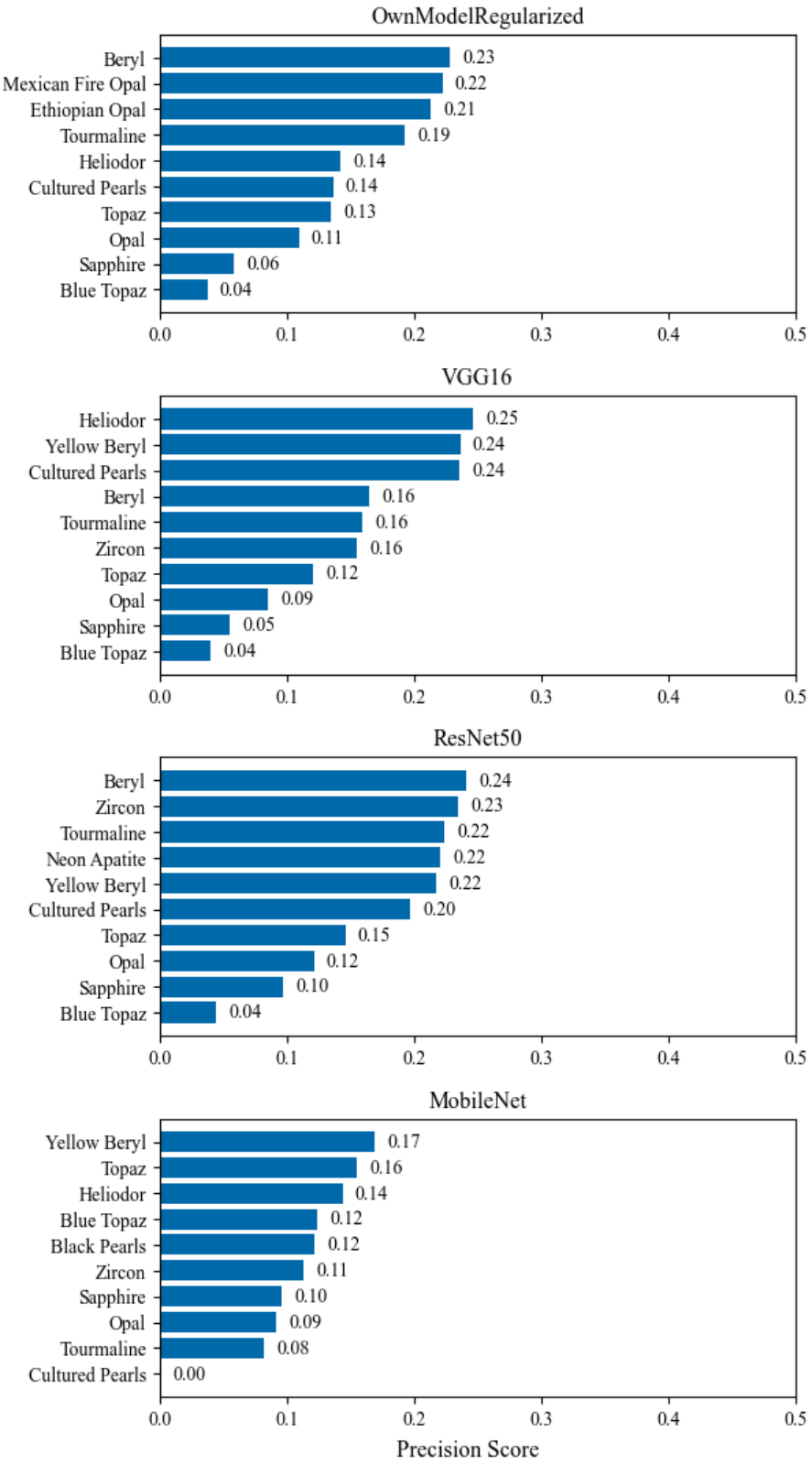


Figure 4.9: Last-10 classes with lowest precision per model for: own model, own model regularized, VGG16, ResNet50, MobileNet.

The precision scores of the ten highest and ten lowest scoring classes are displayed for each model in the Figures 4.8 and 4.9 respectively.

4.3 Other

While the confusion matrix is too large to be properly displayed with almost a hundred classes, it was used to look at the classes with the most misclassifications, which can be seen in Table 4.1.

Missclassified Gemstone	Most %	Most as	Second %	Second as
Blue Topaz	71.0%	Swiss Blue Topaz	7.9%	London Blue Topaz
Sapphire	30.1%	Natural Sapphire	22.3%	Blue Sapphire (Neelam)
Opal	43.9%	Australian Opal	17.4%	Black Opal
Topaz	57.9%	Swiss Blue Topaz	4.6%	London Blue Topaz
Cultured Pearls	72.7%	South Sea Pearls	15.1%	Pearl (Moti)
Heliodor	42.4%	Beryl	40.4%	Yellow Beryl
Tourmaline	27.7%	Green Tourmaline (Verdelite)	9.5%	Alexandrite
Ethiopian Opal	48.5%	Black Opal	16.7%	Opal
Mexican Fire Opal	34.4%	Fire Opal	23.9%	Citrine (Sunela)
Beryl	53.0%	Yellow Beryl	17.2%	Citrine (Sunela)

Table 4.1: Most missclassified gemstones and the gemstones they were most and second most classified as. Extracted from the confusion matrix.

As there is no prediction of a bounding box, IoU is not an applicable evaluation metric.

Similarly, recall is not used since there is no object detection that could be failed.

5 Discussion

5.1 Created Model

After some testing it was evident that D2k outperformed D1k on all models. Reaffirming that more training data leads to better results. Resulting from that only D2k was used for the remainder of the project (Figure 4.1).

The first created model which did not use regularization, performed well with good minimization of the training loss and increase in accuracy leading to fast convergence, as visible in Figure 4.2. However, as previously mentioned in the method section, the model displayed signs of overfitting. Specifically, the validation accuracy stalled and the validation loss increased, while the training loss kept decreasing, even during the fine-tuning epochs with lowered learning rate. Based on these findings the regularisation was implemented, notably improving the model as shown in Figure 4.3.

Considering solely the validation accuracy, after 20 epochs an accuracy of 0.54 is achieved (Figure 4.4). Taking into account that the classification is considering almost a hundred classes, many of which share similarities, which will be further discussed later on, this is a reasonable performance. With a top-3 accuracy of 0.85 and a top-5 accuracy at 0.91, the model can confidently provide a shortlist of the most likely gemstone candidates. This could be of great use with the model offering assistance in narrowing down potential matches and making informed decisions without having to conduct several tests on them.

5.2 Comparison to state of the art models

While VGG16 appears to converge well, there seems to be an indication of overfitting similar to the first version of the own model (Figure 4.5 (a) and (c)).

In comparison, ResNet50 does not show signs of overfitting and its performance improves the most of all the models from the fine-tuning epochs (Figure 4.5). The peak in loss after the first fine-tuning epoch is not concerning, considering the parameters of 50 layers were all changed at once after being unfrozen.

MobileNet shows an even stronger increase of validation loss than the first iteration of the own model did (Figure 4.5 (a) and (e)). This indicates even stronger overfitting. Equally to ResNet50, a peak of loss occurs at the beginning of fine-tuning. However, the accuracy first drops but recovers to its level before the fine-tuning but not exceeding it.

Out of all models, ResNet50 had the lowest accuracy, top-3 and top-5 accuracy before fine-tuning and showed the biggest improvements outperforming the other models (Figure

4.6 and 4.7). Before fine-tuning the developed model with regularization performed best in all accuracies but it only showed a slight increase in performance during fine-tuning. For the top-5 validation accuracy the new model only differed by 0.01 to the performance of ResNet50.

To take a closer look at the performance of gemstone classes one at a time the precision score was considered. While the top-10 precisions indicate great identification of the classes shown in Figure 4.8, the bottom-10 scores in Figure 4.9 are very low for all of the models. There are only minor performance differences between the models, with VGG16 and ResNet50 performing the best, followed by the regularized own model and then MobileNet.

A further investigation into the cause of the missclassifications causing the low precisions was conducted by looking up the most missclassified gemstones and the stones there were missclassified as. The most common causes of missclassification were found to be intra class variations and inter class similarities.

Figure 5.1: Example of intra class similarities for Sapphires.

Intra class variations lead to missclassifications for classes with many internal differences. An example of this can be seen in Figure 5.1 for sapphire with some of its many color variations.

Figure 5.2: Example of inter class similarities.

Furthermore, many inter class similarities could be noted, an example of which can be

seen in Figure 5.2. When considering the findings of most prominent misclassifications from the confusion matrix, as listed in Table 4.1, there seems to be an overlap not only of related classes but especially for gemstones of the same kind that are sourced from different regions.

5.3 Conclusion

In conclusion, while the newly created model was only best performing before fine-tuning, it performed almost as well as ResNet50 after fine-tuning. Considering the differences in complexity, the created lightweight model was able to keep up well with state of the art models.

Validation accuracy alone was between 0.4 and 0.6 for all of the models. Since a breaking down of overall possible gemstones was also considered as goal for this project, top-3 and top-5 accuricies which include the further predictions of the model were compared. These accuricies improved vastly, showing sucessful suggestions of possible gemstones.
For the missclassifications that did occur, two main causes could be found with intra class variations and inter class similarities, specifically the sub division of classes by regionality.

Future enhancements of the model itself could include residual connections to improve towards the performance of ResNet50 or including additional information like the Mohs-Scale of gemstones, which describes their hardness. To tackle the difficulty in distinguishing gemstones by regionality a second network could be implemented. As for the training data, microscopic pictures of gemstones might lead to performance increases as well since they could likely better capture internal structures required for classification. This could also include adding images of raw gems. Assuming such data could be gathered a powerful network that might even be able to distinguish natural and synthetic gemstones might be possible. This could be further advanced through a value estimation of the gemstones. Overall, while this project was successful in creating a lightweight model able to break down a hundred possible gemstone classes into three to five options, there are still many options to delve deeper into gemstone identification.

6 References

Adrian Rosebrock. (2022). *Intersection over Union (IoU) for object detection.* Retrieved March 7, 2024, from https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

Ceylons Munich. (2023). *How to recognize genuine gemstones?* Retrieved March 10, 2024, from https://www.ceylons.de/blog-en/recognize-genuine-gemstones.

Cloud Factory. (2024). *Accuracy score.* Retrieved March 7, 2024, from https://wiki.cloudfactory.com/docs/mp-wiki/metrics/accuracy.

Consumer Technology Association. (2021). *The AI Pastry Scanner that's Fighting Cancer.* Retrieved March 10, 2024, from https://www.ces.tech/articles/2021/may/the-ai-pastry-scanner-thats-now-fighting-cancer.aspx.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2009). L2 regularization for learning kernels. In (p. 109–116). AUAI Press.

Daria Chemkaeva. (2020). *Gemstones Images.* Retrieved March 3, 2024, from https://www.kaggle.com/datasets/lsind18/gemstones-images.

Evidently AI. (2024). *Accuracy vs. precision vs. recall in machine learning: what's the difference?* Retrieved March 6, 2024, from https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall.

Geodes. (2024). *Gem Sorting.* Retrieved March 9, 2024, from https://www.gemporia.com/en-gb/learning-library/terms/gem%20sorting/.

He, K., Zhang, X., Ren, S., & Sun, J. (2016, 06). Deep residual learning for image recognition. , 770-778. doi: 10.1109/CVPR.2016.90

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017, 04). Mobilenets: Efficient convolutional neural networks for mobile vision applications.

International Gem Society. (2024a). *Gemstone Encyclopedia.* Retrieved March 9, 2024, from https://www.gemsociety.org/gemstone-encyclopedia/ .

International Gem Society. (2024b). *An Introduction to Gem Identification.* Retrieved March 9, 2024, from https://www.gemsociety.org/article/how-gems-are-identified/ .

Jeweller Pawnbroker. (2024). *Cuttings.* Retrieved March 9, 2024, from https://cuttingsjewellers.co.uk/blog/how-many-types-gemstones-are-there .

Päivi Lujala. (2009). *GEMDATA.* Retrieved March 3, 2024, from http://www.paivilujala.com/gemdata.html.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014, jan). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn.*

Res., *15*(1), 1929–1958.

Technische Universität Ilmenau. (2024). *Flora Incognita.* Retrieved March 10, 2024, from https://floraincognita.de/.

The Gemmological Association of Great Britain. (2022). *Beginner's Guide: Everyday Gemstone Identification in the Field.* Retrieved March 9, 2024, from https://gem-a.com/beginner-s-guide-everyday-gemstone-identification-in-the-field/ .

Wikipedia. (2023). *Canny edge detector.* Retrieved March 6, 2024, from https://en.wikipedia.org/wiki/Canny_edge_detector.

Wikipedia. (2024a). *Convolutional Neural Network.* Retrieved March 6, 2024, from https://de.wikipedia.org/wiki/Convolutional_Neural_Network.

Wikipedia. (2024b). *Neural network (machine learning).* Retrieved March 6, 2024, from https://en.wikipedia.org/wiki/Neural_network_(machine_learning).