# Distant Seeing: Applying Machine Vision Algorithms to Historical Scientific Images

## Human features representation in textbook illustrations

**MA Semester Project**

**Laboratory for the History of Science and Technology**

**Spring 2023**


**Professor:** Dr. Jérôme Baudry
**Supervisor:** Semion Sidorenko
**Supervisor:** Dr. Ion-Gabriel Mihailescu
**Student:** Naël Dillenboug

# Contents

**Abstract**

In this paper, we tested numerous models made for object detection before concluding that fine-tuning a model was a necessary step towards the analysis of features in physics books. We compared various parameters to fine-tune to the best of our ability a model capable of extracting features from a large dataset of illustrations. We find that optimal performances should be reached by a model using a Unet++ architecture, with a resnext101_32x16d encoder, using Instagram encoder weights, a batch size of 32 and a distribution of illustrations containing features between 50% and 33% according to our tests. We then analysed several books and found no particular trends in the usage of hands throughout time in physics books but did seem to find trends regarding the usage of hands with specific chapters. We also found examples of reused engraving plates with hands as features in illustrations or redrawn illustrations across authors.

# 1 Introduction

Physics as one of the oldest scientific disciplines has many subjects, from optics to classical mechanics. Physics textbooks have played a pivotal role in disseminating this various knowledge throughout the world. However, with this abundant information, there are some challenges in extracting meaningful information efficiently.

Traditional approaches have relied on manual techniques, while these methods are effective, they require a massive workforce, a cost that can only be justified for some key research. This of course inhibits the exploration of many scientific studies.

By making use of object detection technologies, this project aims to analyse a vast amount of data in a relatively short period, in order to study trends and do quantitative and qualitative analysis. For this, we will experiment with several methods of detecting some features and compare the performances of these methods. Furthermore, we will discuss the challenges associated with such tasks, while exploring how this project can open new research directions.

# 2 Goal of the project

This project is meant to extract features from a particular type of data: illustrations in physics books from the 19th to the 20th century in order to analyse these features. We would like to identify, if possible, any trend in feature usage in illustrations, may they be in time, in frequency or even in the usage of the feature in an illustration. This means we first need to identify these features. We then need to verify that our extraction method for these features does not induce too much bias as we want to use our extraction method on different books from different authors and bias in our extraction method would inevitably induce bias in our analysis. We can then finally aggregate our data in order to perform an analysis.

# 3 Dataset

Books from various sources such as *Google books*, *gallica.bnf.fr*, *e-rara.ch* *digitale-sammlungen.de* and *archive.org* are accumulated in order to obtain a large enough dataset that we could analyse. This amounted to 102 books from 4 different languages making a total of 78'810 pages. With books ranging from the 1800s to the 1900s. This period coincided with changes in the printing technology of drawing, as images started being engraved on wooden blocks instead of copper plates attached at the end of the volumes. Our data consist mostly of French books as it is argued that physics books emerged as a scientific genre in France in the second half of the 19th century. All the books we used in this project are listed in the annexe under subsection 9.2

## 3.1 Features selection

After a first look into the dataset, these features were the most common ones[1]:

[1] All illustrations used here and throughout this report were extracted from the books listen in subsection 9.2

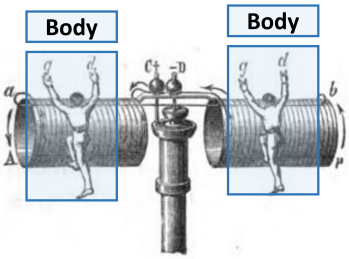


Figure 1: A body used to represent the flow of the current

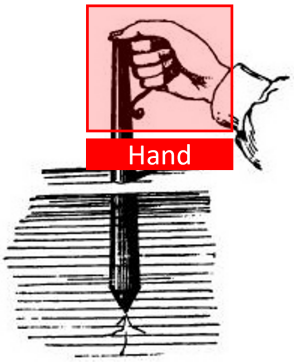


Figure 2: A hand using a tool

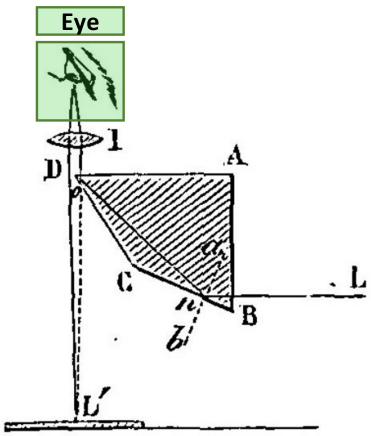


Figure 3: An eye, used to demonstrate optical path

Taking a subset of 7442 images from various books, we found:

- 216 hands
- 49 eyes
- 5 body

The problem of the few illustrations containing eyes is amplified by the fact that we find two representations that seem to have similar usage but have vastly different representations.

Figure 4: Eye representation (external)

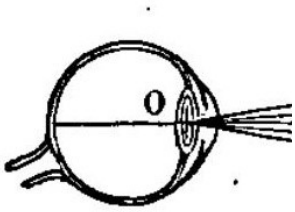


Figure 5: Eye representation (internal)

This further divides the number of eyes into two different representations that a model would need to learn.

The usages of these features are various, where hands a used to mostly demonstrate the usage of tools and to point at objects, and eyes are mostly used in optics chapters to demonstrate optical paths. Bodies seem to be used to demonstrate the flow of current in some books.

From this, we decided to focus our attention on hands as these were the most frequent features. Because of time constraints, we were not able to train models for other features.

# 4 Extracting features

In order to extract features, we wish to find, build or fine-tune a machine-learning model that could accurately predict the existence of a feature in an illustration.

## 4.1 Prexisting models

**Applying common models:** Two main problems were apparent, firstly, these models were trained on very different data for different functions. They did not possess classes that fit our needs. The training data for these models were mostly pictures from datasets such as COCO[LMB+14]. So they were neither of the particular features we are interested in: hands nor were they represented in the same way as our illustrations: sketches or engraving printing.

We wondered whether a hand would be classified as a body since it may be the closest label to a hand available for these models but some quick tests demonstrated that it would not be classified as such reliably.

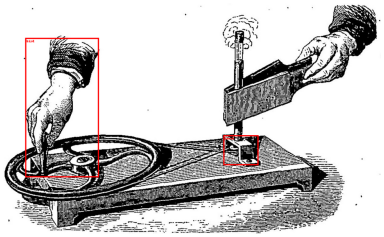

Figure 6: Predictions from fasterrcnn_resnet50_pfn_v2 [Pyta]

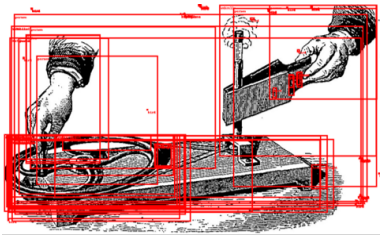

Figure 7: Predictions from retinanet_resnet50_fpn [Pytc]

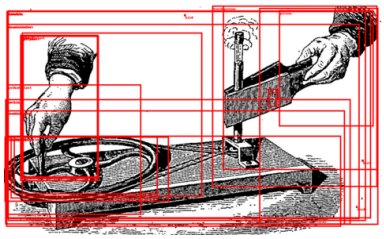

Figure 8: Predictions from fcos_resnet50_fpn [Pytb]

**Models build on datasets of pictures of hands:** Observing the performance of pre-existing projects was a necessary step in our process, as we wanted to compare with other methods. For this reason, we used the project "Hands Segmentation in PyTorch - A Plug and Play Model" [Cam21]. This project uses several datasets to build a model from DeepLabV3 with ResNet50 in order to recognise hands.

- **EgoHands** : with 4800 labelled frames, this dataset is composed of pictures taken from a first-person point of view. This is pretty distinct from our project. [BLCY15]
- **EgoYouTubeHands** : with 774 labelled frames, this dataset is also composed of pictures taken from a first-person point of view, sourced from youtube. [UB18]
- **GTEA** : with 1067 labelled frames, this dataset also uses pictures taken from a first-person point of view. [FRR11] [LYR15]
- **Hand Over Face** : This dataset is the first to provide labelled pictures not taken from a first-person point of view. This dataset is however very small with only 180 labelled frames. [UB18]

Attempting to predict hands from such a model was a failure as the gap between sketches/engraving printing and pictures was too wide and most of the dataset were pictures of hands taken from an angle that is not found commonly in our application as we can compare figures 9 and 10.

Figure 9: Prediction from a picture

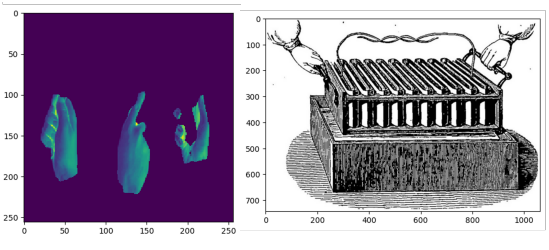

Figure 10: Predictions from an illustration

## 4.2 Building a model

As not to spend the entire duration of the project labelling illustrations, we only selected a few images, which meant our labelled dataset was of modest size. Building a model from scratch was not an option with our dataset size, fine-tuning one was however a better approach. We fine-tuned models using different architectures, encoders, encoder weights and hyperparameters, also using different distributions of features in the training set in order to understand their effect on the performance of our model.
As we compare the models, we refer to them by an integer, this is in reference to table 2 where all model performances are summarized. All models are trained until performances plateau, we then select the best model that does not display signs of overfitting.

**Labelling:** In order to build a training set, a validation set and a testing set, we first needed to label enough data manually in order to have a comprehensive dataset. For this, we used the Universal data tool [UDT], allowing us to annotate over 7442 images.

Our dataset has 7229 images containing no hands, 155 images containing one hand, 52 containing two hands and only 6 containing three hands as plotted in figure 11.

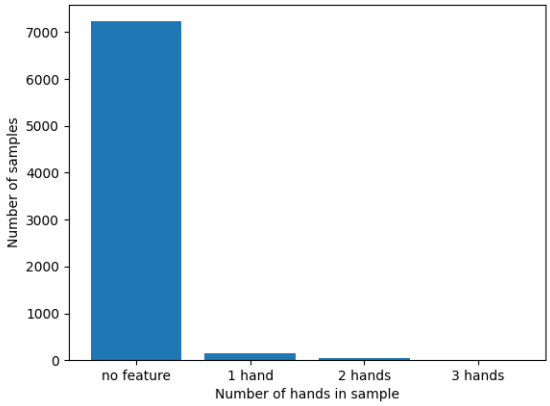


Figure 11: Distribution of hands our labelled dataset

From this step, we extract a JSON file. This allowed us to have a lightweight and versatile way of comparing predictions with ground truths, compared to converting our labels to mask images as it is often done.
We always use a data split between the training set, the testing set, and the validation set of 80%, 10%, 10% in terms of illustrations containing features.
Additionally, only after having already trained a few models with a shuffled training/validation and testing set, did we sense the need to have an additional testing dataset of illustrations that were never used by any trained models in order to compare all datasets in a fair way. For this, we decided to label 3'000 additional illustrations.

**Choices made in labelling:** During labelling, some samples are quite ambiguous, for this reason, only labels with realistic human hands that were an important part of the picture were labelled in the dataset. This means we tried not to label images that were part of the "body" (figure 1), as these would be found by searching for other kinds of features.

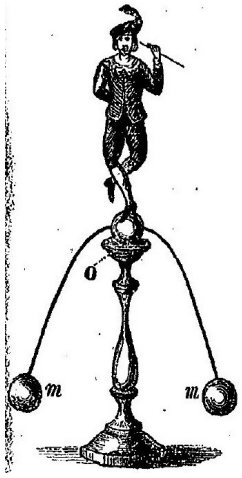


Figure 12: Hand not an important part of the image

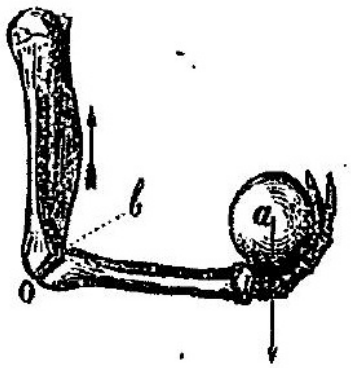


Figure 13: Not a realistic hand

Here we do not label a skeleton hand (figure 13) as a hand, as this is too rare of a hand representation in our dataset and will only add confusion. We also do not label the hand in figure 12, as the hand is too small and does not serve the same usage as most hands in our samples.

### 4.3 Extracting all sketches from books

As we are mostly concerned about sketches/engraving printing in physics notebooks, we want our model to only be trained with such illustrations and the data on which we apply our model to only be illustrations contrary to entire pages. For this, we used a tool created previously by a student who worked on the same project with the same datatype and made a script extracting the illustrations from each page. [Met22]

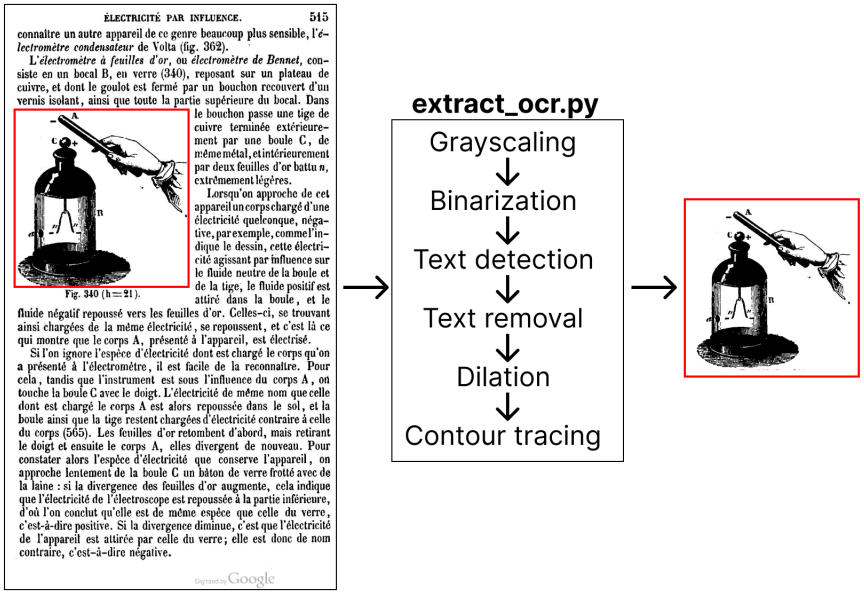


Figure 14: Extraction process from extract_ocr.py

**Results from extraction:** With 104 books that amount to 78'810 pages, we extracted 166'295 images. Some of the extracted illustrations are due to inaccuracy in the extracting script. We find mostly two kinds of errors after extraction :

Figure 15: Gutter between pages classified as an illustration (rotated 90°)

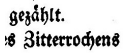


Figure 16: Text classified as an illustration

The first type of error seen in figure 15 comes from under-cropped pages during the scan of these books. Something that we have little control over as simply removing lines would also remove an entire sub-genre of illustrations, mostly used in optics. The second type of error seen in figure 16 probably comes from the font used by this particular author (*MULLER*), the OCR used in the algorithm is not used to fraktur calligraphy. This is also true of some formulas in books, as these are hard for OCR to detect.

These miss-classified images do represent a sizeable amount of data that we could filter out if need be but since our model should detect images with some predetermined features, these images should be easily classified as not possessing any features and not hinder any analysis.

## 4.4 Metric used

**Validation set used for metric computation** As we want to compare several models, we explored the metrics that could be useful for such a task. Before addressing how we will compute this metric, we must determine what data we want to compute the metrics from. In order for this metric to translate as best as possible the actual performance of the model, we should use the same distribution of features in the data as in our application. We also should use data never used before for training any of the models that we wish to compare, as we discussed in paragraph 4.2.

Once a testing dataset is made, it is necessary to determine a threshold separating positive from negative values. This process is explained in paragraph 4.4 and 4.4.

**Thresholding:** Any metrics necessitate a threshold in order to determine what positive and negative predictions are. As we have only one class, this choice is binary.

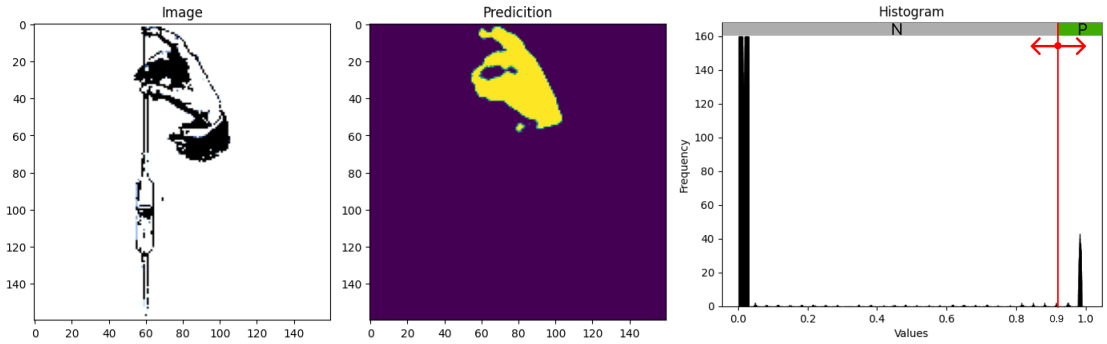


Figure 17: Thresholding

As our predictions have very strong confidence (near 100%). We apply a threshold of 90%, this choice is explained further in paragraph 4.4. As we can see here, the threshold has very little effect on true positive (TP), false positive (FP), false negative (FN) and true negative (TN) as long as we are not using a value too close to 0 or 1. We explain these metrics below.

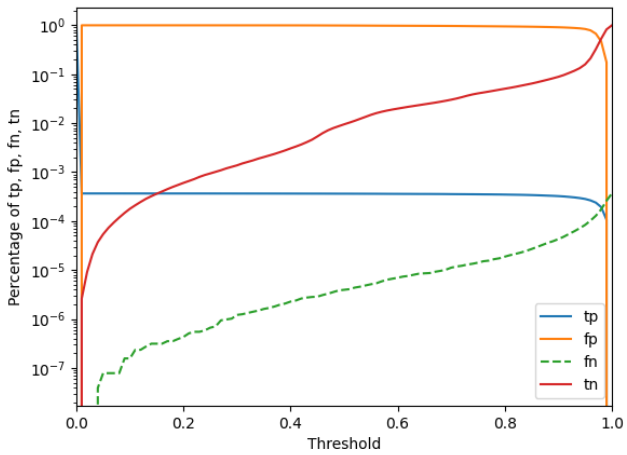


Figure 18: TP, FP, FN, TN for our worst performing model (model 29)

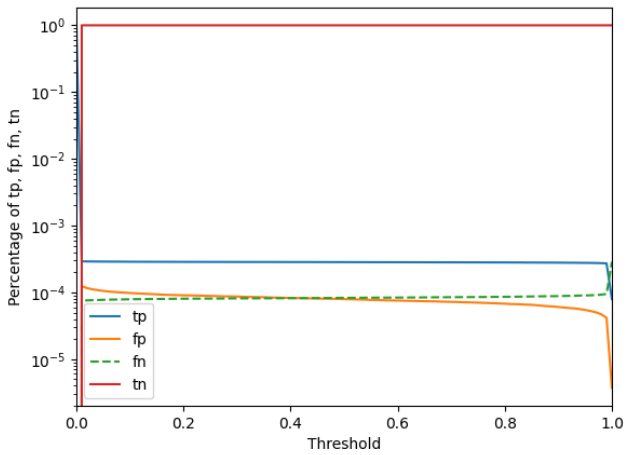


Figure 19: TP, FP, FN, TN for a good performing model (model 38)

**TP, FP, FN, TN:** These are the basis of every performance metric we are using, they are dependent on the threshold as we defined in paragraph 4.4. As we are only training using one class, this is our case.

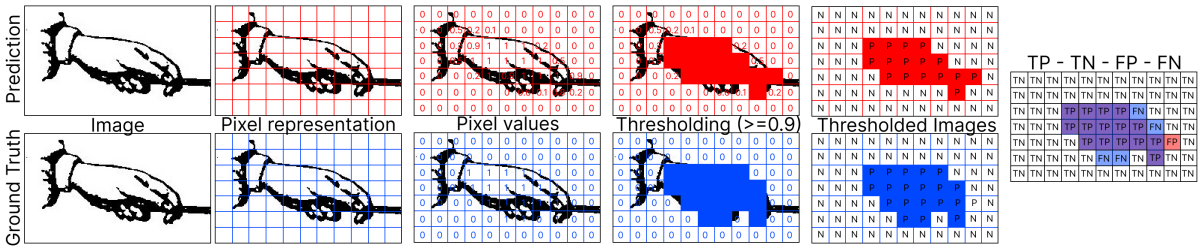


Figure 20: Explanation of how TP, FP, FN and TN are computed

In figure 20, we oversized the pixel grid in order to explain how TP, FP, FN and TN are computed. By using the ground truth and a thresholded prediction, we obtain positive and negative values for the prediction and ground truth. We can then compare these two and determine if the prediction is true or false.

- TP: A predicted positive pixel with a positive value on the ground truth for the same pixel represents a true positive.
- FP: A predicted positive pixel with a negative value on the ground truth for the same pixel represents a false positive.
- FN: A predicted negative pixel with a positive value on the ground truth for the same pixel represents a false negative.
- TN: A predicted negative pixel with a negative value on the ground truth for the same pixel represents a true negative.

We will as of now use TP, FP, FN, and TN to refer to these values for readability.

**Imagewise vs. Datasetwise:** Any metrics can be measured image-wise and dataset-wise, this does make a difference particularly when working with such a large class imbalance.
In an image-wise metric, we sum all true positive, false positive, false negative, and true negative pixels over all images and then compute the score of a particular metric $f$.
In a datasets-wise metric, we sum true positive, false positive, false negative, and true negative pixels for each image, then we compute the score of a particular metric $f$ for each image and average the scores over the whole dataset.

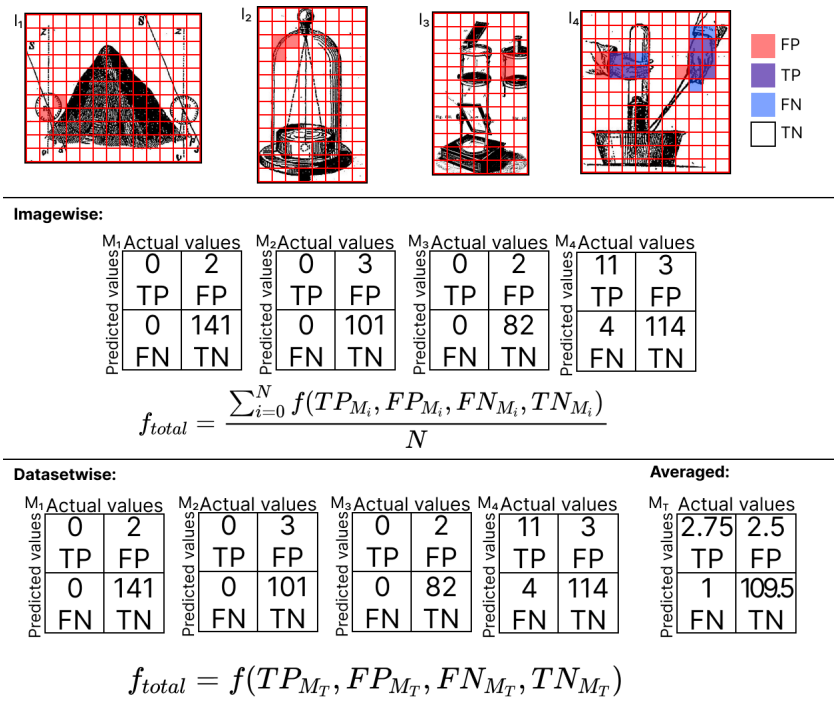


Figure 21: Explanation of Image-wise vs Dataset-wise metrics computation for a metric $f$

In this explanation, we make use of confusions matrices [Pea96], a useful way of representing TP, FP, FN and TN.

**IoU:** The first important metric we need to use in object recognition is Intersection over Union (IoU) metric. This metric can sometimes be used to determine what predictions are true positive and false positive based on a threshold minimal IoU threshold.
The IoU score, also called the Jaccard index also uses the same notion of intersection over union to quantify performance, this is the way we will mostly make use of the metric.

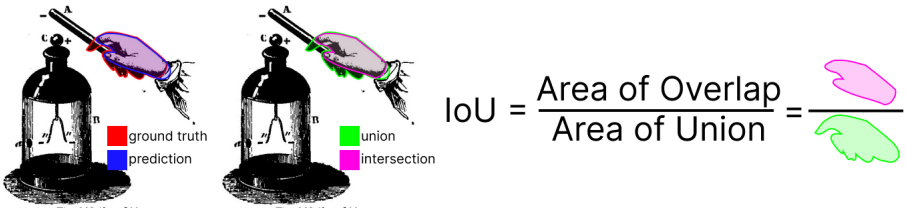


Figure 22: IoU score

The IoU metric is sometimes calculated for different sizes of predictions, this is useful to compare performance on small/medium and large objects. Such a case will be discussed in figure 26.

**Accuracy:** This metric is often used to measure how well a model is performing.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

By calculating the accuracy this way, we actually compare the total number of correct predictions with the total number of predictions.[Kor]

With a large class imbalance in our testing set, this metric is a poor choice in comparing models' performance as the metrics will be very high, leaving not much room for comparison.

**Precision:** This metric allows us to gauge what proportion of positive identifications were actually correct. [Deva]

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

In our application, we care about precision, as we wish to use our model to filter through images and output only images containing features that we care for.

**Recall:** This metric allows us to gauge what proportion of actual positives were identified correctly. [Deva]

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

With features appearing with low frequencies as we observed, we need our application to also archive high recall so as not to miss too many features.

**Precision-Recall plot:** A precision-recall plot combines the data acquired by computing the recall and precision for a large number of thresholds in order to attempt to obtain an optimal threshold.

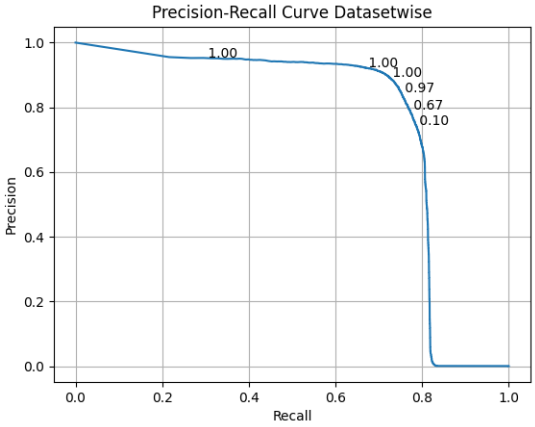


Figure 23: Dataset-wise computed precision and recall plot for model 40.

In figure 23, we do see that our best model ca not reach a recall above 0.8, this does not mean that only 80% of image containing a feature is detected. Since this precision and recall are computed dataset-wise, this means that we only detect up to 80% of the pixels containing a feature. It is important to keep this in mind as we will make repeated use of such plots.
Computing this metric in an image-wise array, we obtain a recall of 0.9969 and a precision of 0.9955 for a threshold of 90%.
As mentioned in paragraph 4.4 and 4.4, we do care about these two metrics. As we archive a good performance for both recall and precision for a threshold of 0.9, this is the threshold we will use unless stated otherwise.
In figure 23 we also see some numbers along the curve, these are the thresholds for the corresponding precision and recall. This is useful information to see how much a threshold is influenced our metric. As we can see from this figure, the values of precision and recall have little variations for thresholding going from 1.0 to 0.1.

**F1 score:** This score is particularly interesting for measuring performance on data with an uneven distribution. This is particularly our case, as we mentioned that our features are rare in our dataset.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

This metric, ranging from 0 to 1 measures accuracy by taking the harmonic mean of the precision and recall. [Wik23a]

**True positive rate:** This metric is synonym to the recall as describe in 4.4.

$$TPR = Recall = \frac{TP}{TP + FN} \tag{5}$$

**False positive rate:** This metric allows us to gauge what is the proportion of false positive over all negative values. It is also the probability of falsely rejecting the null hypothesis. [Wik23b].

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

As with the true positive rate (paragraph 4.4), these values can be computed image-wise or dataset-wise as explained in paragraph 4.4.

**ROC Curve** : Similarly to the Precision-Recall curve described in paragraph 4.4, this curve describes the performance of a model using the true positive rate and the false positive rate (as explained above in paragraphs 4.4, 4.4), for a number of thresholds.

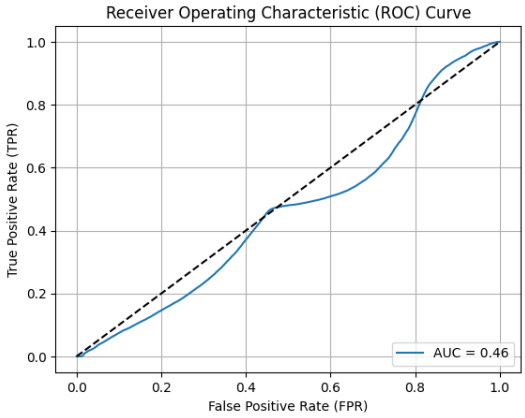


Figure 24: ROC curve for model 29 (with poor performance)

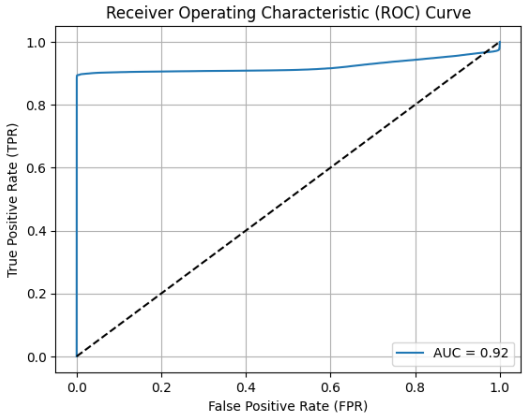


Figure 25: ROC curve for model 42 (with good performance)

These two curves are computed dataset-wise from FPR and TPR.

The line, crossing from (0,0) to (1,1) represents the performance of a model that picks positive and negative values entirely due to chance. This means that our worst performing model (here in figure 24) performs worst, most of the time, then a model picking values entirely due to chance.

**AUC:** As we can observe in figure 24 and 25, we can use the AUC metric with a ROC curve. This metric stands for "Area Under the ROC Curve", integrating the surface bellows the curve gives us a metric that is threshold-invariant.[Devb]

In models with a high threshold performance variance, this is not a well-fitting metric but as we will see in paragraph 4.4, most of our trained models that perform quite well, are not very threshold dependant.

**Metrics used:** We will make use mostly of the IoU, F1 and other metrics dependent on the pertinence of the metric while comparing the performances of models. If we do not specify, we use a dataset-wise computed metric, as the values tend to be less capped at 1.0 or 0.0. As mentioned previously, we also used a threshold to sort positive from negative values of 0.9 unless stated otherwise.

## 4.5 Bounding Box

Building a model using bounding boxes means for every image, a label is created with a rectangle the size of our feature. Additional random transforms are added to increase the size of our training set. This method however performed poorly.

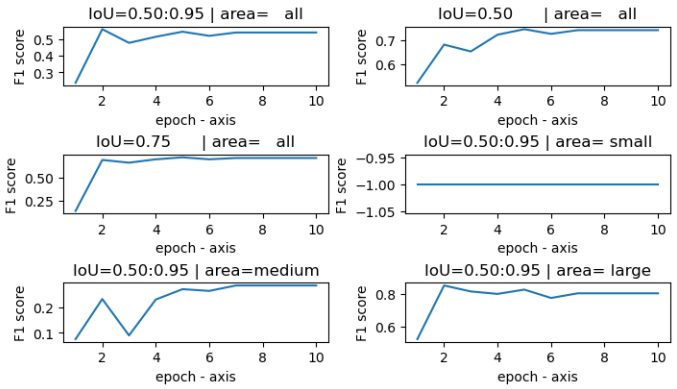


Figure 26: Performance for IoU score of a Finetunned FasterRCNN model using only images of hands

In figure 26, we separated the performance for different IoU thresholds (here used to determine positive and negative predictions) and area. This is done in order to compare the performance on small, medium, large or all objects. With $IoU = 0.5 : 0.95$ meaning the performance is averaged over several thresholds between 0.5 and 0.95.

With such a model, we observed a good performance for the recall but very low precision. We hypothesised that this could come from the usage of a bounding box, as a lot of hands hold objects, and this could add complexity to our task. For this reason, we switch to using segmentation models.

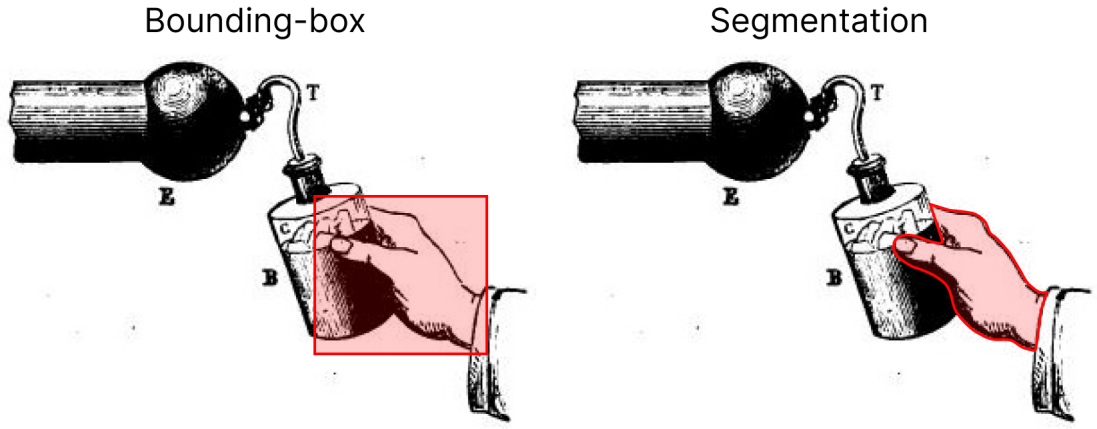


Figure 27: Difference between using a Bounding Box and Segmentation

## 4.6 Segmentation

As stated in subsection 4.5, because many of the features are used in conjunction with objects, we realised that using segmentation may be beneficial, as we could make sure that the masks would not

include these objects. This did require relabelling illustrations with features. Something that is time-consuming but was revealed to be worthwhile.
We will now compare, performance-wise, choices that can be made while fine-tuning for such an application.
We made use of a Python library with Neural Networks for Image Segmentation based on PyTorch called Segmentation Models [Iak19].

**Choice of architecture:** Several architectures were compared in order to find the best-performing architecture:

- Model 44[2]: U-Net, this architecture is meant to be usable with small datasets, something we are obviously interested in as mentioned in 4.2. [RFB15]
- Model 38: U-Net Plus Plus, this architecture, originally designed for medical image segmentation, should enhance the performances of U-Net. [ZSTL18]
- Model 45: Feature Pyramid Network (FPN) [KHGD17] is a feature extractor designed with a feature pyramid concept with accuracy and speed in mind. [Hui]
- Model 46: PAN, an architecture with a fast convergence rate.[MLW19]
- Model 47: DeepLab V3+, this architecture is normally meant for semantic segmentation. [CZP+18]
- Model 48: MANet is a Multi-Scale attention network originally designed for Liver and tumor Segmentation. [FWLW20]

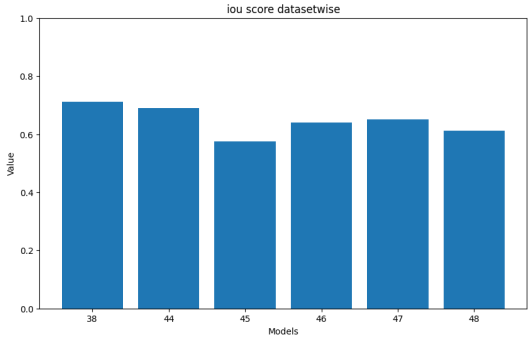


Figure 28: IoU score for models 44, 38, 45, 46, 47, 48

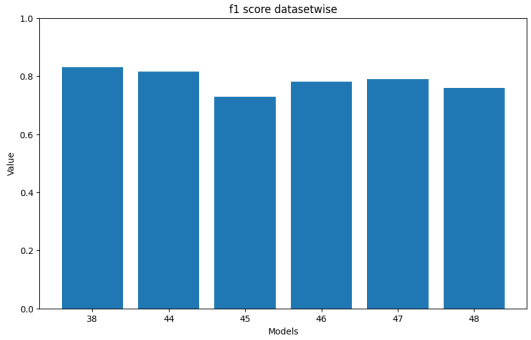


Figure 29: F1 score for models 44, 38, 45, 46, 47, 48

The results from figures 28 and 29 show a slightly better performing Unet architecture while model 45 that used an FPN architecture has the worst performance of the compared architectures, with a $\Delta_{F1} = F1_{Unet} - F1_{FPN} = 0.086144$, almost a 10% decrease in performance.

[2]All model numbers found in this report refer to table 2

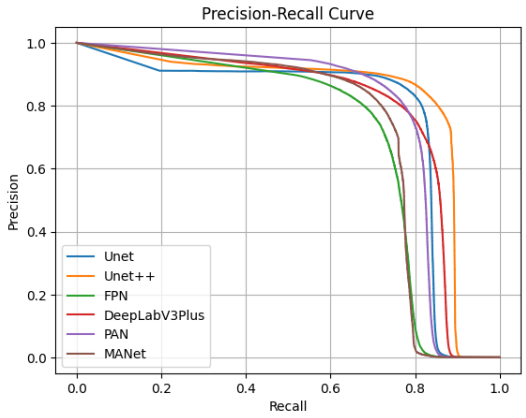


Figure 30: Precision recall curve for models 44, 38, 45, 46, 47, 48

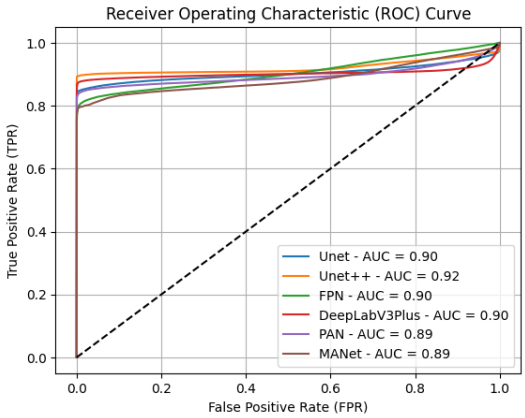


Figure 31: ROC curve for models 44, 38, 45, 46, 47, 48

To get a complete picture, we plot figures 30 and 31 and do see that model 38 with a Unet++ architecture performs slightly better than model 44 using an Unet architecture as this model gets closer to the upper right corner. We also see a better AUC value for Unet++ by about 2%.

**Choice of encoder** ResNeXt : We mostly used this encoder in our work and in particular resnext101_32x8d and resnext101_32x16d. We [XGD+16]

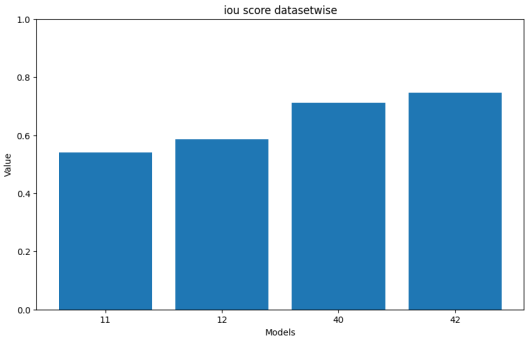


Figure 32: IoU score for models 11, 12, 40 and 42 trained with resnext101_32x8d and resnext101_32x16d

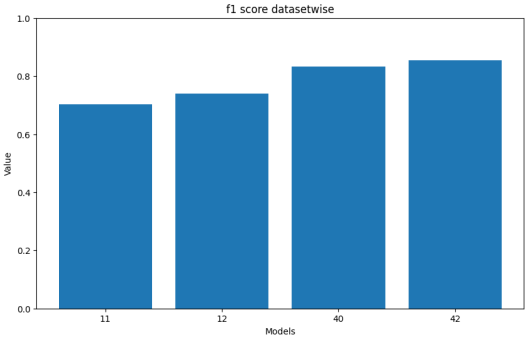


Figure 33: F1 score for models 11, 12, 40 and 42 trained with resnext101_32x8d and resnext101_32x16d

Models 11 and 40 were trained with resnext101_32x8d while models 12 and 42 were trained with resnext101_32x16d. Performance-wise, resnext101_32x16d seems to perform better than resnext101_32x8d but this comes with a heavy computational cost, it is memory and time expensive. For this reason, we mostly used resnext101_32x8d in our tests but any of our performance should only be enhanced using resnext101_32x16d or even resnext101_32x48d.

**Choice of encoder weights** Training from the right weights should improve performance significantly, the weighted computed from an Instagram dataset using ResNeXt seem to perform quite well on sketches object detection. [Wig19] For this reason, we mostly used these weights while training our models.

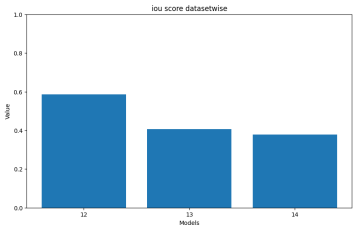


Figure 34: IoU score for 3 models trained with different encoder weights

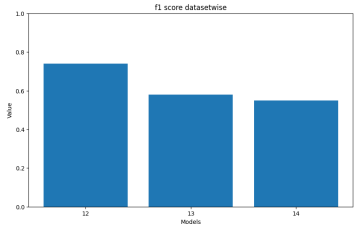


Figure 35: F1 score for 3 models trained with different encoder weights

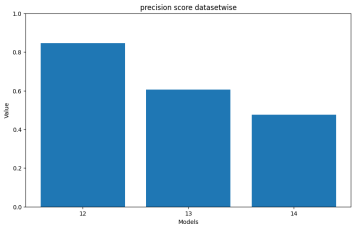


Figure 36: Precision score for 3 models trained with different encoder weights

Figures 34, 35 and 36 taken from models 12, 13 and 14 show that using similar encoders and hyper-parameters, we perform unequivocally better with the Instagram encoder weight even with a slightly worst encoder.

- Model 12 uses a resnext101_32x8d encoder with Instagram weights.
- Model 13 uses a resnext101_32x16d encoder with imagenet weights.
- Model 14 uses a resnext101_32x16d encoder with a semi-supervised learning imagenet model [YJC+19].

**Choice of Loss function** Dice Loss: The Dice Loss is based on the Dice coefficient. This loss was introduced to be used in highly unbalanced segmentation models[SLV+17], this loss fits our needs.

$$DiceLoss = 1 - 2 * \frac{y \bigcap y_{pref}}{y + y_{pref}} [MOK] \tag{7}$$

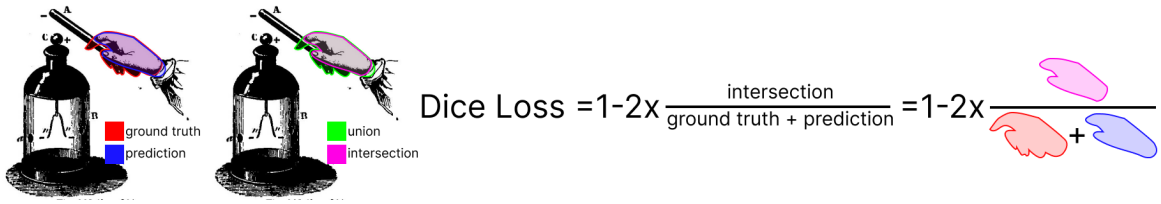


Figure 37: Dice Loss calculation explanation

The Jaccard Loss is based on the IoU calculation detailed in paragraph 4.4.

$$JaccardLoss = 1 - \frac{y_{pref} \bigcap y}{y_{pref} \bigcup y} [RW16] \tag{8}$$

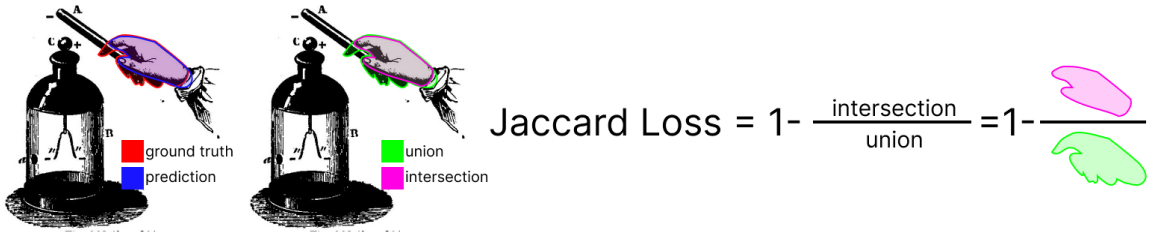


Figure 38: Jaccard Loss calculation explanation

Tversky Loss [SEG17] has a large advantage over the Jaccard and the Dice loss: we can use the parameters $\alpha$ and $\beta$ to compare how much we care about the false positive and false negative.

$$TverskyLoss = 1 - \frac{y_{pref} \bigcap y}{y_{pref} \bigcap y + \alpha * (y_{pref} \setminus y) + \beta * (y \setminus y_{pred})} \text{[SEG17]} \tag{9}$$


Figure 39: Tversky Loss calculation explanation

This should, in theory, allow us to finetune how much we care about FP and FN. However, this method yielded poor results with a low IoU score and a poor f1 score with whatever $\alpha$ and $\beta$ we used. We did not investigate further and went on to find other methods for balancing training detailed in subsection 4.7.

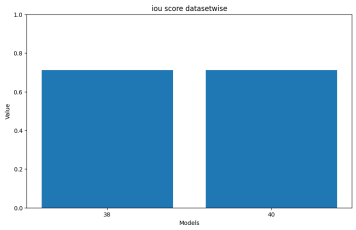


Figure 40: IoU score models 38 and 40 trained with a Jaccard loss and a Dice loss

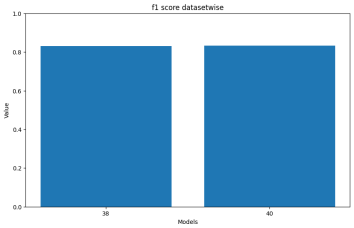


Figure 41: F1 score for models 38 and 40 trained with a Jaccard loss and a Dice loss

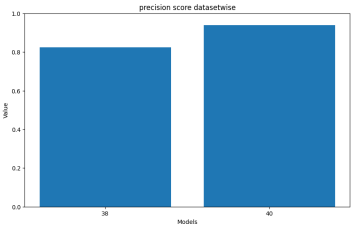


Figure 42: Precision score for models 38 and 40 trained with a Jaccard loss and a Dice loss

As we can see on figures 40 and 41 we have very similar performances for our F1 score and IoU score, however, we do see a gap in performance in figure 42 for the precision.

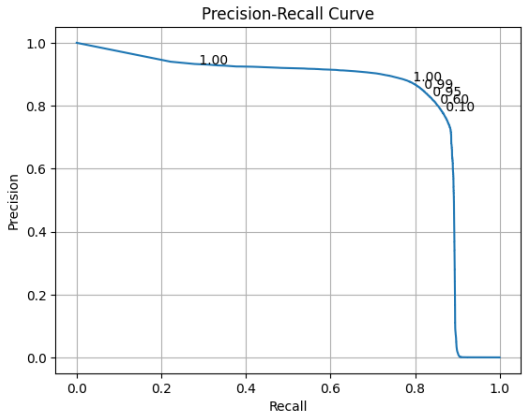


Figure 43: Precision-Recall curve for model 38 trained with a Jaccard loss

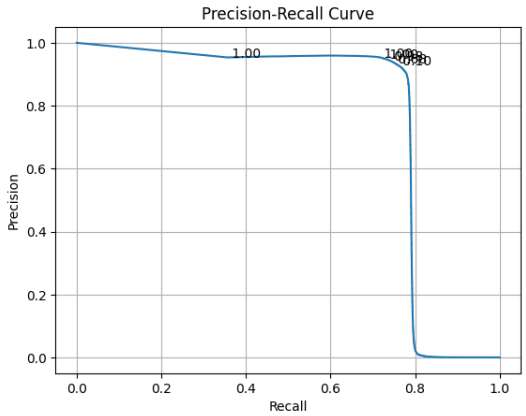


Figure 44: Precision-Recall curve for model 40 trained with a Dice loss

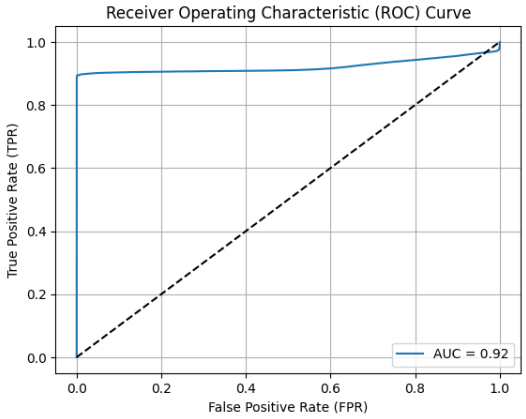


Figure 45: ROC curve for model 38 trained with a Jaccard loss

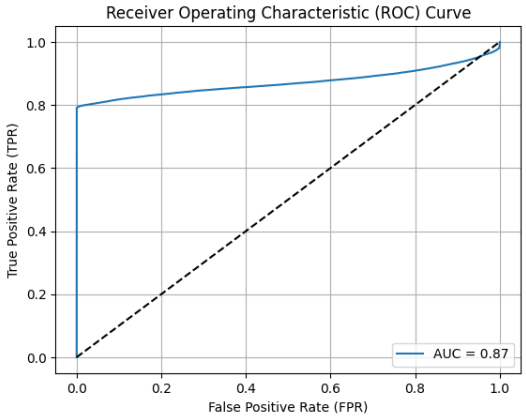


Figure 46: ROC curve for model 40 trained with a Dice loss

Figures 43, 44, 45 and 46 differentiate a bit better the performances using a Jaccard or a Dice loss. As we see, while the model trained with Jaccard loss such as model 38 archives on average a higher recall but a lower precision than model 40 trained with a Dice loss. With a slightly higher AUC value for model 38, we prefer using a Jaccard loss and will use this loss function to further compare training methods.

## 4.7 Datasplit and distribution

**Training with only samples containing features:** In the first phase of the project, we trained our model with images containing the features we wanted to detect. This performed quite well but when applied on images that do not contain hands (usually about 97-99% of images in our application do not contain any features), we end up detecting a large number of false positives. We hypothesised that this was because of class imbalance.

**Getting around class imbalance:** Several ways are available in order to better train a model suffering from class imbalance. we first tried to train our models with the same distribution of features as in our application. This means about 1-3% of illustrations containing hands for example.

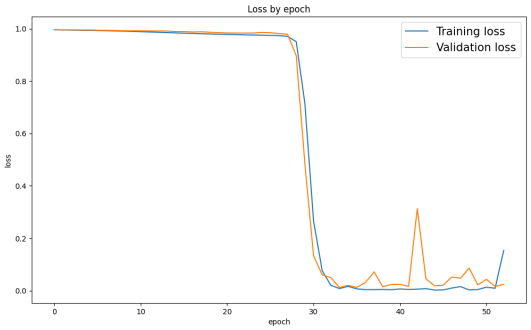


Figure 47: Loss vs epoch plot for model 29 (trained using an illustration with the same distribution of features as our application)

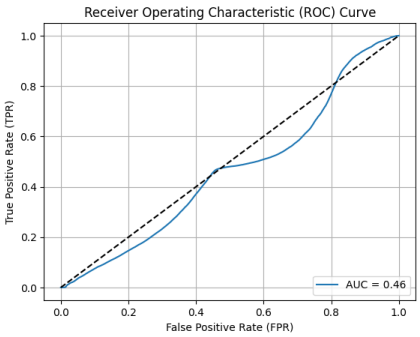


Figure 48: ROC curve of model 29 (trained using an illustration with the same distribution of features as our application)

While training this model, we quickly reach a point after about 28 epochs, where the loss for both training and validation drops quickly. We hypothesised that this is because the model quickly start predicting almost only negative as it was not worth it predicting positive values in a dataset containing such a small ratio of actual positives. This model then performs very badly with an f1 score of 0 and an AUC of 0.46.

Another way we explored correcting our model for class imbalance was of course the usage of various loss functions. This was already covered in paragraph 4.6.

We then went on to explore other methods, one such method was training using a two-stages fine-tuning method as introduced in this paper [VSW+22] but instead of using two class-balanced reweighted loss function, we first train with a class-balanced dataset and then with a class-unbalanced dataset. This also performed quite poorly, with no significant improvements from only training with a class-balanced dataset. Due to time constraints, we could not further investigate this method.

After these tests, we concentrated on testing the effect on our metrics of the ratio of images with no features in the training set.

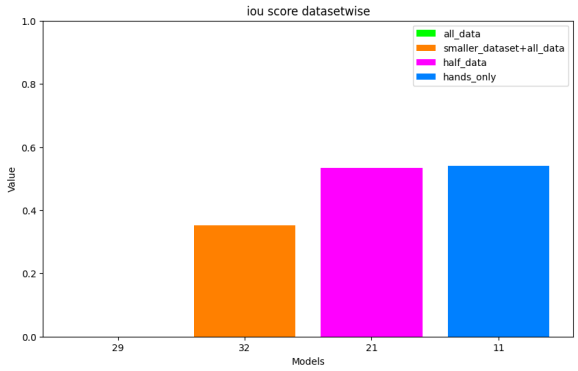


Figure 49: IoU score for 4 models trained with different features distribution in the training set

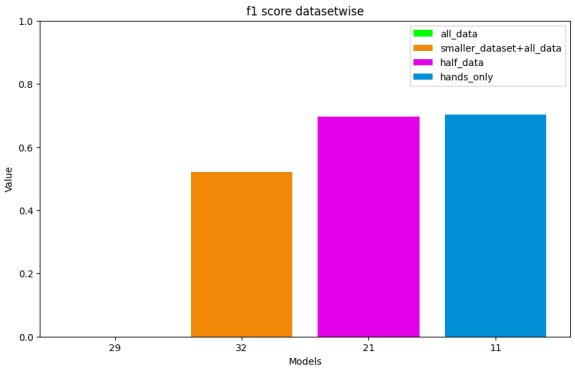


Figure 50: F1 score for 4 models trained with different features distribution in the training set

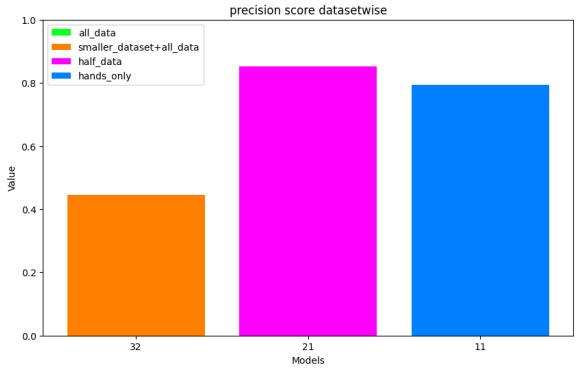


Figure 51: Precision score for 4 models trained with different features distribution in the training set (Precision of model 29 not plotted as the value is 0)

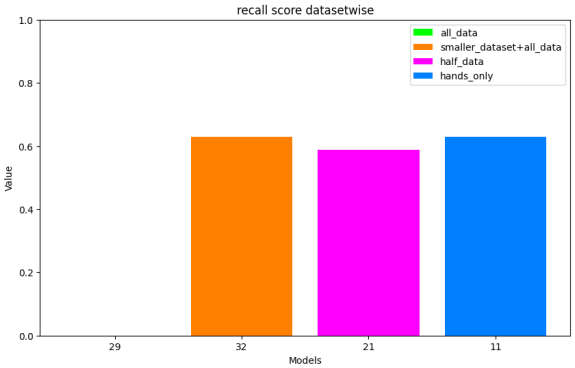


Figure 52: Recall score for 4 models trained with different features distribution in the training set (Precision of model 29 not plotted as the value is 0)

In figures 49, 50 and 51, the following models are compared:

- Model 29 : The training set contains a 3% distribution of images containing features.
- Model 32 : The training was first done with a balanced dataset and then with a dataset with a 3% distribution of images containing features.
- Model 21 : The training set contained as many images with features as without.
- Model 11 : The training set contained only images containing features.

The architecture used was UnetPlusPlus, with a resnext101_32x8d encoder, Instagram encoder weights and a Dice Loss function.

As we see in these figures, models with a too-large class imbalance tend to perform poorly, however, we also see that when we train using only images containing features, we archive a higher f1 and IoU score, however, we have the worst precision, meaning we will have more false positive using this model.

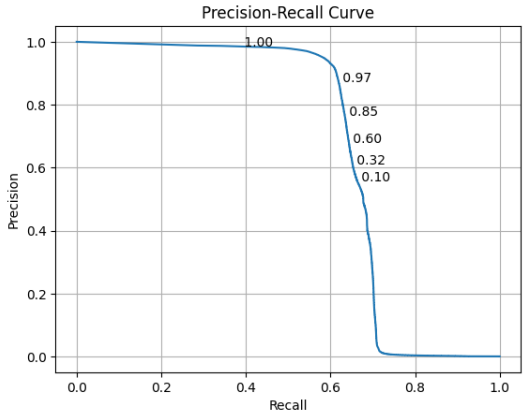


Figure 53: Model 11 dataset-wise Precision-Recall curve

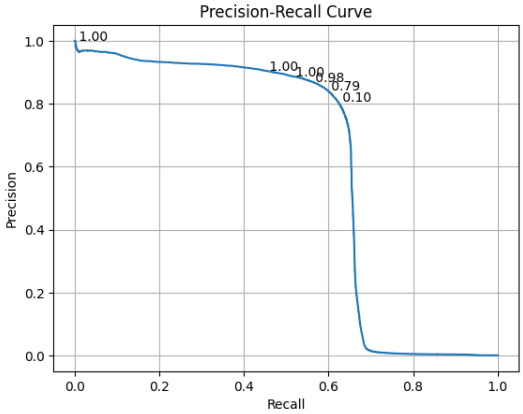


Figure 54: Model 21 dataset-wise Precision-Recall curve

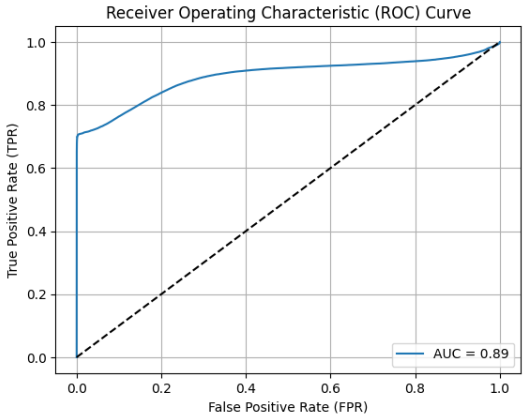


Figure 55: Model 11 dataset-wise ROC curve

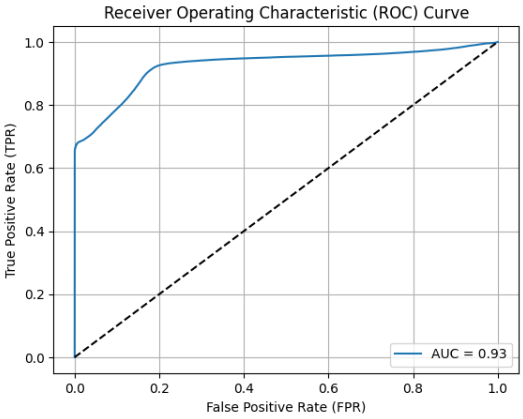


Figure 56: Model 21 dataset-wise ROC curve

Figures 53, 54, 55 and 56 allow us to further understand performance difference between training with a dataset with as many images containing features as without and with a dataset containing only. We do observe that our model is improved by adding images without features in the training set.

**Further balancing the training dataset** In paragraph 4.7, we discussed mitigating the effects of class imbalance, we investigated further this problem by comparing the performance of models with different distributions of illustrations with features in the dataset.
Because we did not want to remove an already small number of illustrations with a feature, we simply added illustrations that did not contain any features. This also means that the results from this paragraph are biased as the training dataset is overall larger for models with a low distribution of illustration-containing features.

- Model 41 : training dataset with around 10% of illustration containing features.
- Model 39 : training dataset with around 33% of illustrations containing features.
- Model 38 : training dataset with around 50% of illustrations containing features.
- Model 43 : training dataset with around 80% of illustrations containing features.

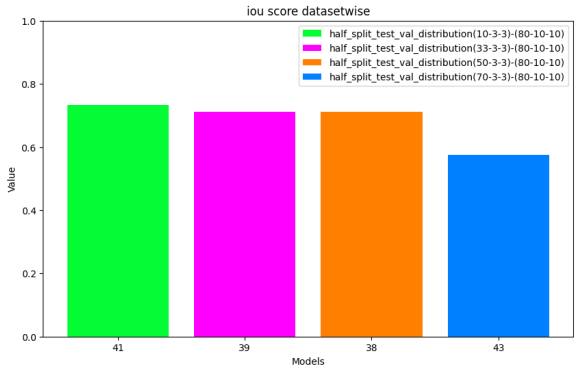


Figure 57: IoU score for 4 models trained with different features distribution in the training set

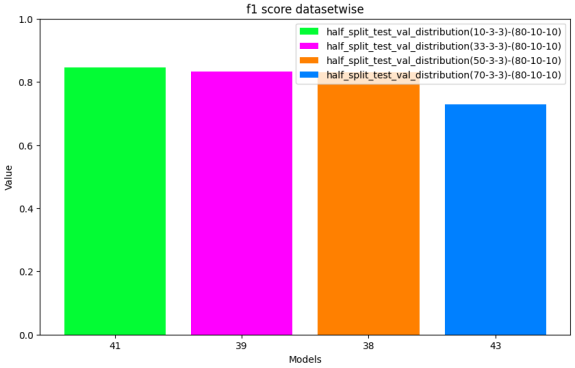


Figure 58: F1 score for 4 models trained with different features distribution in the training set

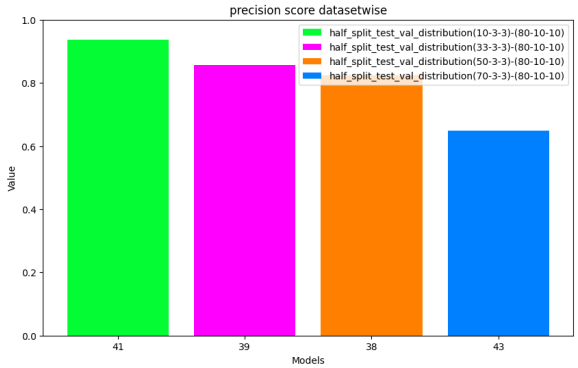


Figure 59: Precision score for 4 models trained with different features distribution in the training set (Precision of model 29 not plotted as the value is 0)

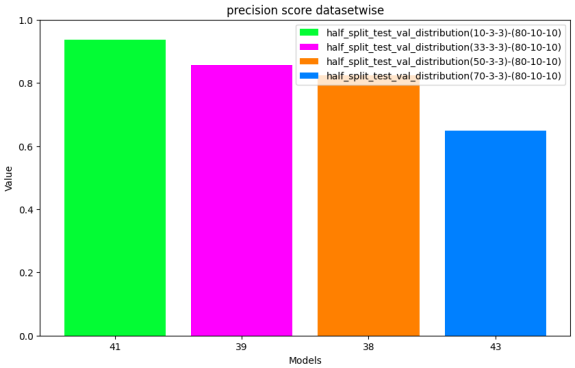


Figure 60: Precision score for 4 models trained with different features distribution in the training set (Precision of model 29 not plotted as the value is 0)

While model 41 or 39, with the least number of illustrations with features proportionally to the dataset, meaning a closer resemblance to the actual distribution of illustrations with features in our application, seem to be performing better (in terms of IoU and f1 score in figures 57 and 58), we also plot the precision-recall and ROC curve to get a better idea of the performance in figures 61 and 62.

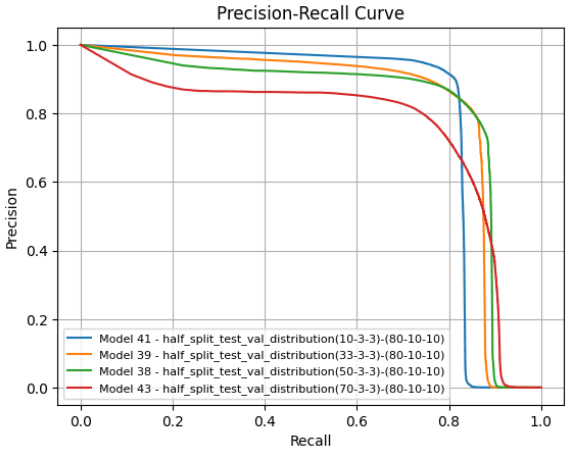


Figure 61: Precision-Recall curve for 4 models trained with different features distribution in the training set

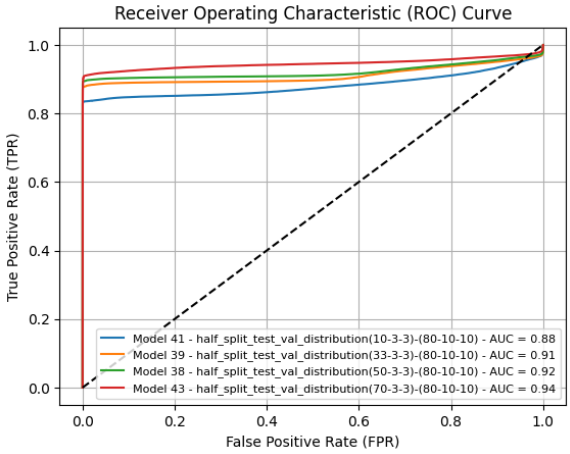


Figure 62: ROC curve for 4 models trained with different features distribution in the training set

With all of this information, our conclusion is a bit different, as we see model 43 seems to have the best AUC score, as it can reach a high recall, but at the cost of a lot of precision. However, models 38 and 39 seem to be a safe bet as they reach a good precision and recall, getting near the upper right corner of the precision-recall plot.

**Stochastic Gradient Descent vs Minibatch Gradient Descent** We started training in stochastic gradient descent, which means that our batch size was equal to one image.[Bro20] We improved performance and computation speed by using "Minibtach gradient descent". For this, we wanted to compare performance with different batch sizes. Changing batch size does reduce computational time but we do need to resize our images as it was to memory intensive to keep high-resolution images with large batch sizes.

In figures 63, 64, 65 and 66 we plotted the performance of models 49 and 38, trained in identical ways, with the illustrations of the same dimensions, except for the batch size which is 16 for model 49 and doubled for model 38.

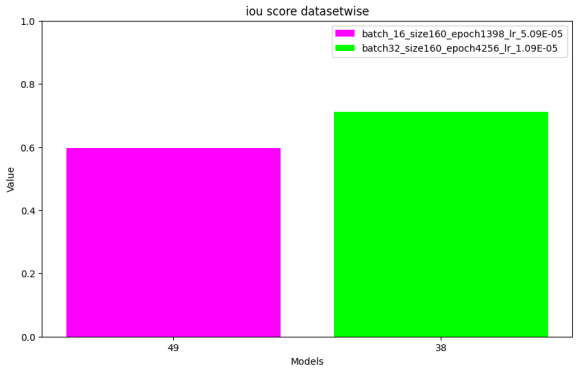


Figure 63: IoU score for models 49 and 38

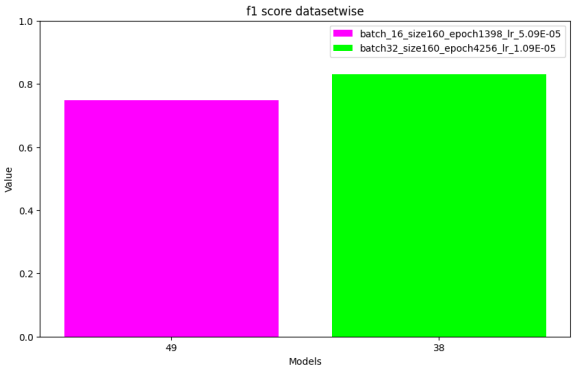


Figure 64: F1 score for models 49 and 38

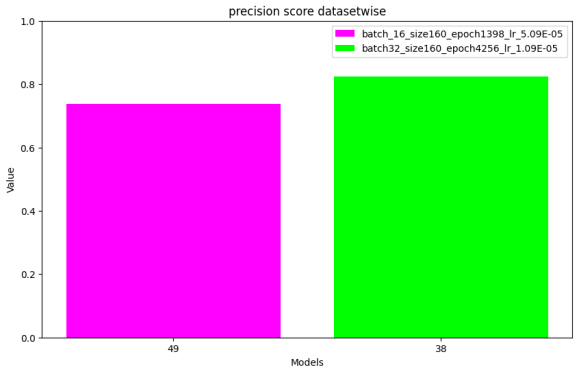


Figure 65: Precision for models 49 and 38

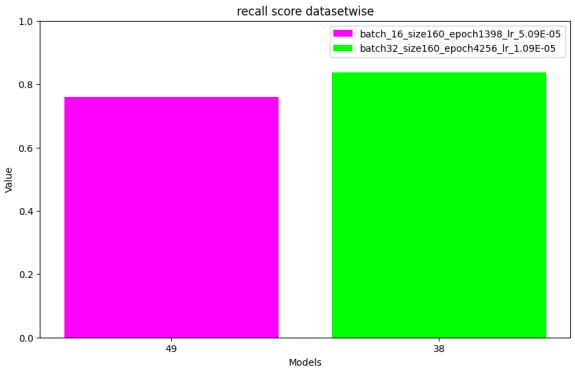


Figure 66: Recall for models 49 and 38

As we see, a larger batch size seems to enhance performance, this could be explained by the fact that it increases the chance of an illustration containing a feature being present in the current batch, allowing the model to better generalise.

**Conclusion of comparison:** As we compared a lot of choices that can be made, the best scenario seems to be using a Unet++ architecture, with a resnext101_32x16d encoder, using Instagram encoder weights, a batch size of 32 and a distribution of illustrations containing features between 50% and 33%. We were however unable to train with this exact configuration due to constraints limiting our computational resources.

# 5 Resulting predictions

**Using a custom confidence calculation:** As we have seen in figure 17, thresholds seem to be very concentrated near 0 or near 1. In order to select good confidence, we could reuse the notion of threshold, which we determined using a Precision-Recall plot as explained in paragraph 4.4. This would simply mean selecting high enough predicted values as detect features such as in figure 67.

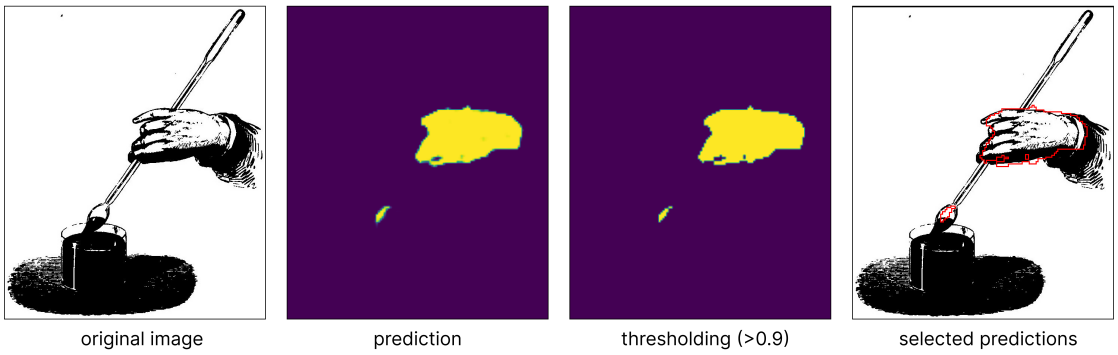


Figure 67: Basis confidence thresholding method

However, we experienced another method of differentiating between good and bad predictions.

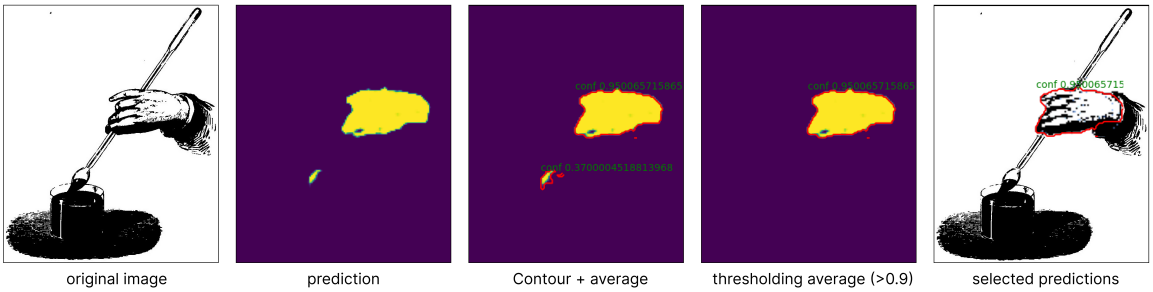


Figure 68: Basis confidence thresholding method

In order to more accurately filter the predictions after training the model, we find the contour of every shape and find the mean value of confidence (i.e. the mean of all the values of the pixel in the confidence map inside the contour). We then threshold the result by the average value of confidence. This means objects that have high confidence for most pixels, will be classified as positive, while objects that do contain some high confidence but are also close to pixels with low or average confidence are classified as negative. As we can see in figure 67 compared to 68 we can remove a false positive by using the second method.

Code Listing 1: Code used to extract confidence

```
results = []
for image in apply_loader.dataset:
    prediction = torch.from_numpy(image).to(DEVICE).unsqueeze(0) #Get prediction
    pr_mask = best_model.predict(prediction)
    pr_mask  = pr_mask.to(DEVICE)
    conf_np = pr_mask[0][0].cpu().numpy() #Build a confidence map
    binary_image = np.uint8(conf_np>0.001)#Turn all values but background to TRUE
    contours, hierarchy = cv2.findContours(binary_image, cv2.RETR_EXTERNAL, cv2.
                                           CHAIN_APPROX_SIMPLE) #Find contour
    all_boxes = []
    all_conf = []
    all_std = []
    if len(contours) > 0:
        for contour in contours:
            mask_tmp = np.zeros_like(binary_image)
            cv2.fillPoly(mask_tmp, pts = [contour], color=255) #Create a mask with
                                                               contour
            mean_value, _, _, _ = cv2.mean(conf_np, mask=mask_tmp) #Compute mean of
                                                               confidence inside of contour
            all_conf.append(mean_value) #Append to all confidence predictions for this
                                                               illustration
```

```
    current_result = {'image':apply_loader.dataset.images_fps[i], 'image_shape':image.
                                                shape,'contour': contours, 'confidence'
                                                 : all_conf}
    results.append(current_result)
```

We used some iteration of code 1 to store the predictions and confidences, this allows us to change the level of confidence we wish to use for a particular analysis.

**Resulting confidence distribution:** With 166'962 images extracted from books, we should have a very large number of images that contain no hands.

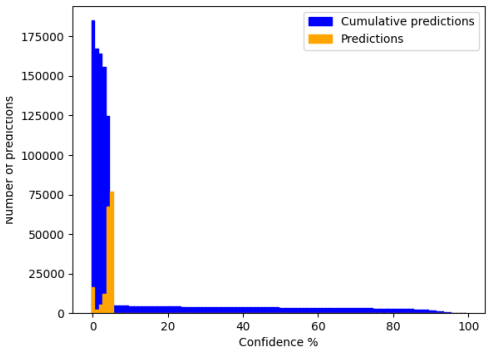


Figure 69: Distribution and cumulative distribution distribution of confidence from 0% to 100%

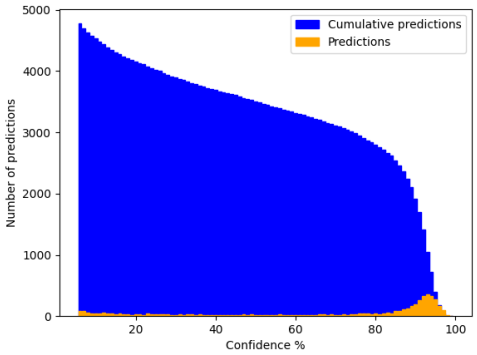


Figure 70: Distribution and cumulative distribution of confidence from 6% to 100%

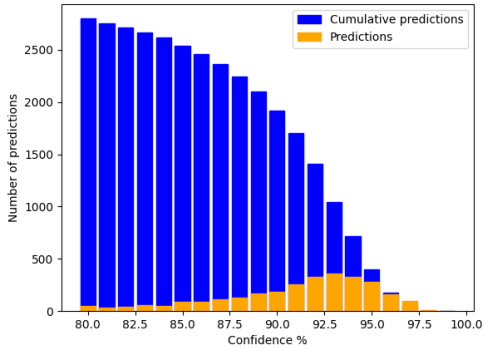


Figure 71: Distribution and cumulative distribution of confidence from 80% to 100%

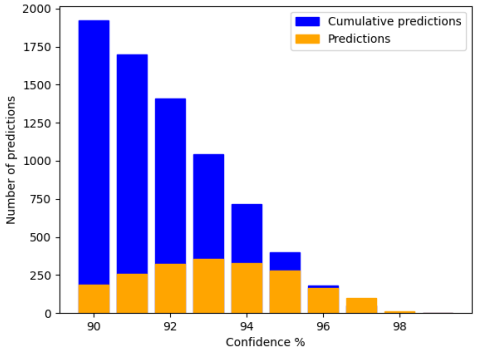


Figure 72: Distribution and cumulative distribution of confidence from 90% to 100%

As we can see in figure 69, we do classify a lot of pictures with very little confidence, between 0% and 6% confidence. This means a vast number of shapes are classified as negative. This would be expected as we expect our features to be rare.

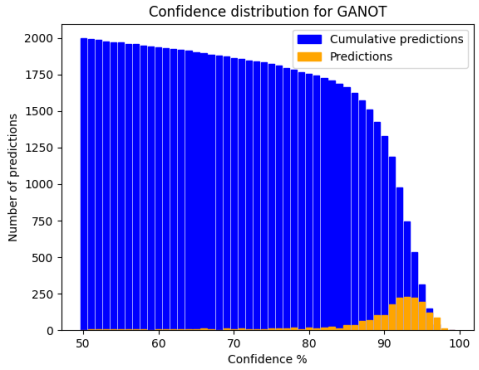


Figure 73: Confidence distribution for illustration found in books from author GANOT

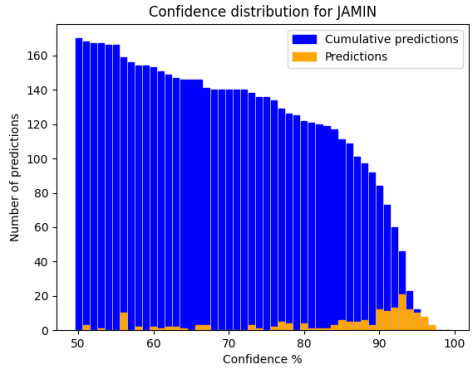


Figure 74: Confidence distribution for illustration found in books from author JAMIN

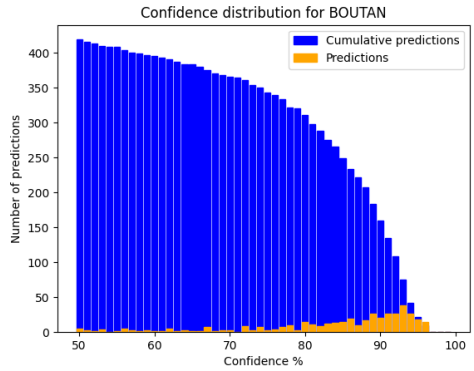


Figure 75: Confidence distribution for illustration found in books from author BOUTAN

However, distributions are not uniform in every book, in figure 73, 74 and 75 we have plotted the distributions from books written by three authors to demonstrate this.
By switching from a confidence level of 90% to 80% on books from author *GANOT*, we get rid of 427 predicted hands meaning we lose about 28% of the data.
With books from author *JAMIN* it is 38 predicted hands that we get rid of, representing 37% of the data from this author.
The choice of confidence level is even more significant for author *BOUTAN*, as we lose 151 predicted hands, representing 64% of the data. This led us to believe we should use a different confidence level for every book/author in order to give accurate results, but due to time constraints, we did not have the chance of finding optimal confidence levels for each book/author. We hypothesised that this was in part because of the absence in the training data of illustrations from some authors, something that would induce a bias towards certain authors that had samples from their illustrations used in our training data.

## 5.1 Analysis

With the data gathered, we can start by analysing the frequency of apparition of our feature in books from different periods and different authors. This data allows us to search for trends.

### 5.1.1 Position of features through the books

A first type of analysis is finding how features are spread out through the books, we would expect to have features more present in specific chapters and not so much in others for example.

For example, hands could be more present in chapters about mechanics and less in chapters about fluids. In order to test this hypothesis, we simply plotted the predictions in every book by page number and will analyse the results for two of them in paragraphs 5.1.1 and 5.1.1.

**Feature position in *DAGUIN, P.A., 1861. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 3, Edition 2*:** In figures 76 and 77, the $x$ axis corresponds to the pages in this specific book, here starting at 0 and end with page 858. The $y$ axis corresponds to the number of illustrations containing at least one hand. We compare figure 76 and 77 to compare results from our model and from a manual search of features throughout the book.

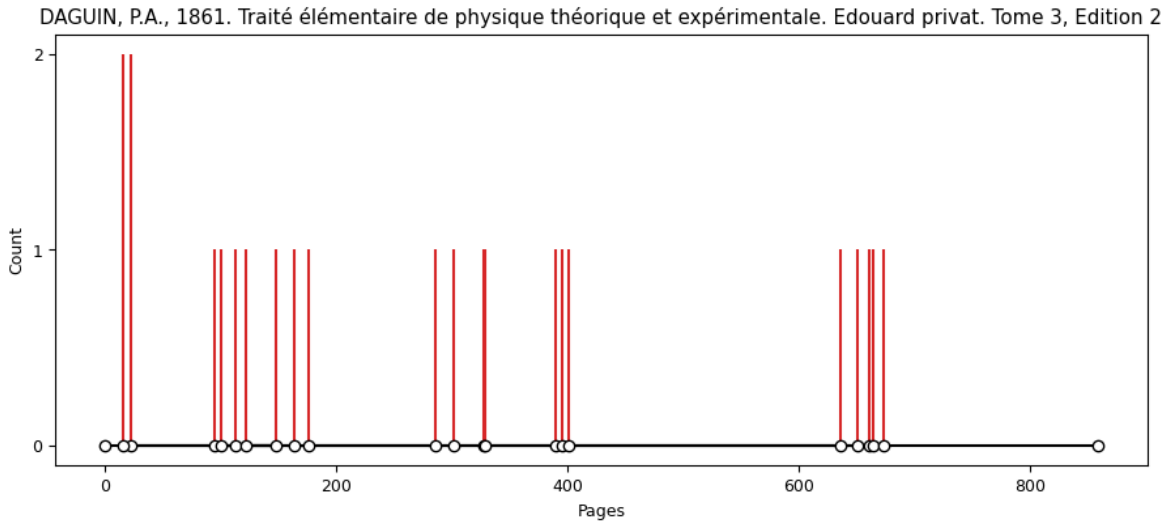


Figure 76: Predicted hands throughout *DAGUIN, P.A., 1861. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 3, Edition 2*

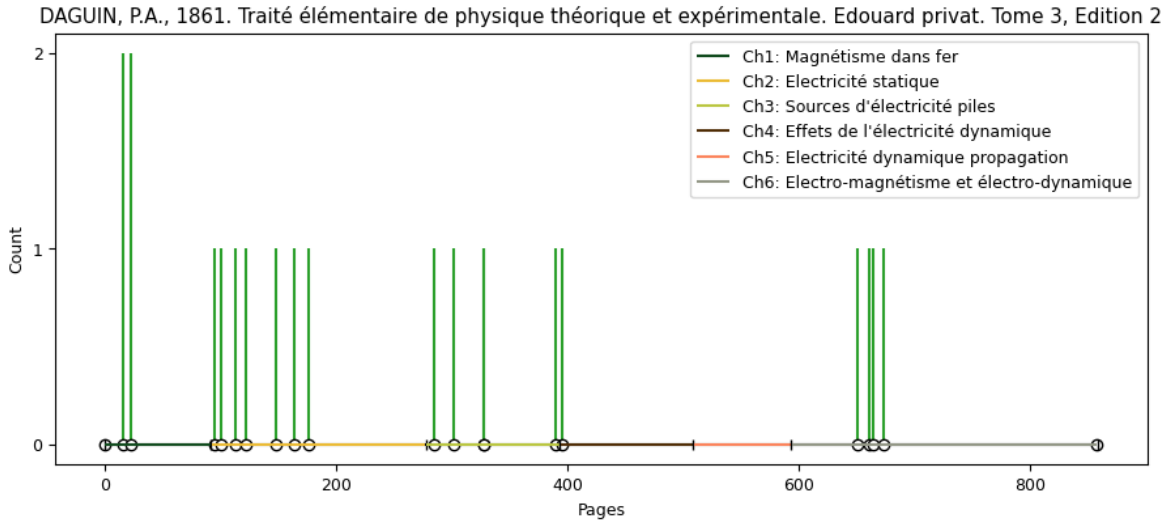


Figure 77: Manually found hands throughout *DAGUIN, P.A., 1861. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 3, Edition 2*

And we can visualise the illustration from *DAGUIN, P.A., 1861. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 3, Edition 2* right here:

Figure 78: FN, p.15

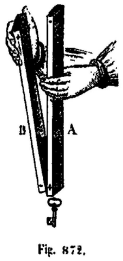

Figure 79: FN, p.15

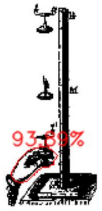

Figure 80: TP, p.22

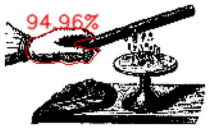

Figure 81: TP, p.95

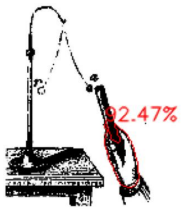

Figure 82: TP, p.100

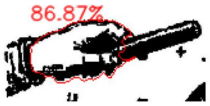

Figure 83: TP, p.113

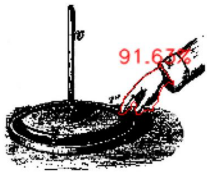

Figure 84: TP, p.122

Figure 85: TP, p.148

Figure 86: TP, p.164

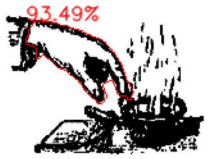

Figure 87: TP, p.176

Figure 88: TP, p.285

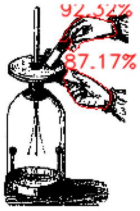

Figure 89: TP, p.301

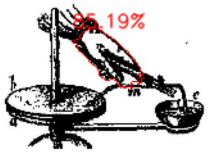

Figure 90: TP, p.327

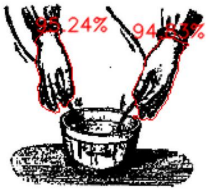

Figure 91: TP, p.328

Figure 92: TP, p.390

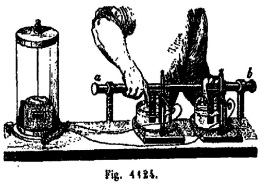

Figure 93: FN, p.395

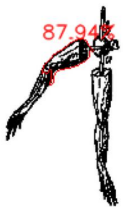

Figure 94: FP, p.401

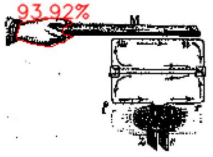

Figure 95: TP, p.651

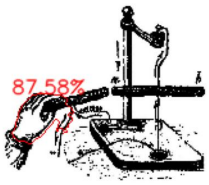


Figure 96: TP, p.661


Figure 97: TP, p.664

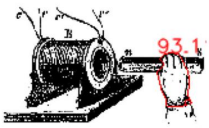


Figure 98: TP, p.673

All these illustrations come from *DAGUIN, P.A., 1861. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 3, Edition 2.* Illustrations are captioned FP, FN and TP for false positive, false negative and true positive. As we can see, figures 78, 79 and 93 were not correctly detected using confidence (as explained in paragraph 5) of 0.85. They are however detected using a custom threshold of 0.8, but this also comes at the cost of adding one false positive.
Figure 94 is a false positive, we would need a confidence level of above 0.88 in order to get rid of the false positive but this would also mean not detecting 83.
Overall, the results are satisfactory as we still get a good understanding of the usage of this feature in this book such as demonstrating an action in an experiment.

Studying the table of content for this particular volume give us some insight into which chapters these features are more used in.

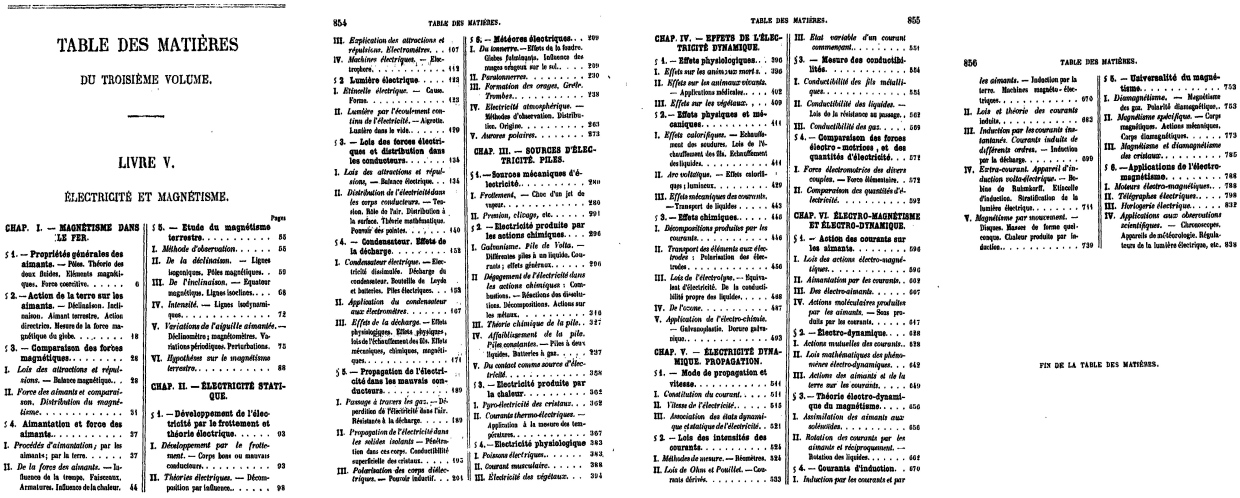

TABLE DES MATIÈRES

DU TROISIÈME VOLUME.

LIVRE V.

ÉLECTRICITÉ ET MAGNÉTISME.

Pages

CHAP. I. — MAGNÉTISME DANS LE FER. 1

§ 1. — Propriétés générales des aimants. — Pôles. Théorie des deux fluides. Aimants magnétiques. Force coercitive. . . . . 6

§ 2. — Action de la terre sur les aimants. — Déclinaison. Inclinaison. Aimant terrestre. Action directrice. Mesure de la force magnétique du globe. . . . . 18

§ 3. — Comparaison des forces magnétiques. . . . . 28

I. *Lois des attractions et répulsions.* — Balance magnétique. . 28

II. *Force des aimants et comparaison. Distribution du magnétisme.* . . . . . 31

§ 4. Aimantation et force des aimants. . . . . 37

I. *Procédés d'aimantation ;* par les aimants ; par la terre. . . . . 37

II. *De la force des aimants.* — Influence de la trempe. Faisceaux. Armatures. Influence de la chaleur. 44

§ 5. — Etude du magnétisme terrestre. . . . . 55

I. *Méthode d'observation.* . . . . . 56

II. *De la déclinaison.* — Lignes isogoniques. Pôles magnétiques. . 59

III. *De l'inclinaison.* — Equateur magnétique. Lignes isoclines. . . 68

IV. *Intensité.* — Lignes isodynamiques. . . . . 72

V. *Variations de l'aiguille aimantée.* — Déclinomètre ; magnétomètres. Variations périodiques. Perturbations. 75

VI. *Hypothèse sur le magnétisme terrestre.* . . . . . 88

CHAP. II. — ÉLECTRICITÉ STATIQUE.

§ 1. — Développement de l'électricité par le frottement et théorie électrique. . . . . 93

I. *Développement par le frottement.* — Corps bons ou mauvais conducteurs. . . . . 93

II. *Théories électriques.* — Décomposition par influence. . . . . 98

854 TABLE DES MATIÈRES.

III. *Explication des attractions et répulsions. Electromètres.* . . 107

IV. *Machines électriques,* — Electrophores. . . . . 113

§ 2. Lumière électrique . . . . 123

I. *Bâtonnelle électrique.* — Cause. Forme. . . . . 123

II. *Lumière par l'écoulement continu de l'électricité.* — Aigrette. Lumière dans le vide. . . . . 129

§ 3. — Lois des forces électriques et distribution dans les conducteurs. . . . . 134

I. *Lois des attractions et répulsions,* — Balance électrique. . 134

II. *Distribution de l'électricité dans les corps conducteurs.* — Tension. Rôle de l'air. Distribution à la surface. Théorie mathématique. Pouvoir des pointes. . . . . 140

§ 4. — Condensateur. Effets de la décharge. . . . . 153

I. *Condensateur électrique.* — Electricité dissimulée. Décharge du condensateur. Bouteille de Leyde et batteries. Plan électriques. . 153

II. *Application du condensateur aux électromètres.* . . . . . 167

III. *Effets de la décharge.* — Effets physiologiques. Effets physiques ; lois de l'échauffement des fils. Effets mécaniques, chimiques, magnétiques. . . . . 171

§ 5. — Propagation de l'électricité dans les mauvais conducteurs. . . . . 189

I. *Passage à travers les gaz.* — Déperdition de l'électricité dans l'air. Résistance à la décharge. . . . 189

II. *Propagation de l'électricité dans les solides isolants* — Pénétration dans ces corps. Conductibilité superficielle des cristaux. . . . . 195

III. *Polarisation des corps diélectriques.* — Pouvoir inducteur. . . 204

§ 6. — Météores électriques. . . 209

I. *Du tonnerre.* — Effets de la foudre. Globes fulminants. Influence des nuages orageux sur le sol. . . . 220

II. *Paratonnerres.* . . . . . 230

III. *Formation des orages, Grêle. Trombes.* . . . . . 238

IV. *Electricité atmosphérique.* — Méthodes d'observation. Distribution. Origine. . . . . 245

V. *Aurores polaires.* . . . . . 273

CHAP. III. — SOURCES D'ÉLECTRICITÉ. PILES.

§ 1. — Sources mécaniques d'électricité. . . . . 280

I. *Frottement,* — Cône d'un jet de vapeur. . . . . 280

II. *Pression, clivage,* etc. . . . . 291

§ 2. — Electricité produite par les actions chimiques. . . . 296

I. *Galvanisme. Pile de Volta.* — Différentes piles à un liquide. Courants ; effets généraux. . . . . 296

II *Dégagement de l'électricité dans les actions chimiques :* Combustions. — Réactions des dissolutions. Décompositions. Actions sur les métaux. . . . . 316

III. *Théorie chimique de la pile.* 327

IV. *Affaiblissement de la pile. Piles constantes.* — Piles à deux liquides. Batteries à gaz. . . . 337

V. *Du contact comme source d'électricité.* . . . . . 358

§ 3. — Electricité produite par la chaleur. . . . . 362

I. *Pyro-électricité des cristaux.* . . 362

II. *Courants thermo-électriques.* — Application à la mesure des températures. . . . . 367

§ 4. — Electricité physiologique 383

I. *Poissons électriques.* . . . . . 383

II. *Courant musculaire.* . . . . . 388

III. *Electricité des végétaux.* . . 394

TABLE DES MATIÈRES. 855

CHAP. IV. — EFFETS DE L'ÉLECTRICITÉ DYNAMIQUE.

§ 1. — Effets physiologiques. . . 396

I. *Effets sur les animaux vivants.* . 396

II. *Effets sur les animaux récents.* — Appareils spéciaux. . . . 402

III. *Effets sur les végétaux.* . . . 409

§ 2. — Effets physiques et mécaniques. . . . . 411

I. *Effets calorifiques.* — Echauffement des soudures. Lois de l'échauffement des fils. Echauffement des liquides. . . . . 411

II. *Arc voltaïque.* — Effets calorifiques ; lumineux. . . . . 429

III. *Effets mécaniques des courants.* — Transport de liquides . . . . 445

§ 3. — Effets chimiques. . . . . 446

I. *Décompositions produites par les courants.* . . . . . 446

II. *Transport des éléments aux électrodes :* Polarisation des électrodes . . . . . 456

III. *Lois de l'électrolyse.* — Equivalent d'électricité. De la conductibilité propre des liquides. . . . 468

IV. *De l'ozone.* . . . . . 487

V. *Application de l'électro-chimie.* — Galvanoplastie. Dorure galvanique. . . . . 493

CHAP. V. — ÉLECTRICITÉ DYNAMIQUE. PROPAGATION.

§ 1. — Mode de propagation et vitesse. . . . . 511

I. *Constitution du courant.* . . . . 511

II. *Vitesse de l'électricité.* . . . . 518

III. *Association des états dynamique et statique de l'électricité.* . 521

§ 2. — Lois des intensités des courants. . . . . 524

I. *Méthodes de mesure.* — Résistance. 524

II. *Lois de Ohm et Pouillet.* — Courants dérivés. . . . . 533

III. *Etat variable d'un courant commençant.* . . . . . 551

§ 3. — Mesure des conductibilités. . . . . 554

I. *Conductibilité des fils métalliques.* . . . . . 554

II. *Conductibilité des liquides.* — Lois de la résistance au passage. 562

III. *Conductibilité des gaz.* . . . . 599

§ 4. — Comparaison des forces électro-motrices, et des quantités d'électricité. . . . 572

I. *Force électromotrice des divers couples.* — Force élémentaire. 572

II. *Comparaison des quantités d'électricité.* . . . . . 592

CHAP. VI. ÉLECTRO-MAGNÉTISME ET ÉLECTRO-DYNAMIQUE.

§ 1. — Action des courants sur les aimants. . . . . 596

I. *Lois des actions électro-magnétiques.* . . . . . 596

II. *Aimantation par les courants.* . 602

III. *Des électro-aimants.* . . . . . 607

IV. *Actions moléculaires produites par les aimants.* — Sens produits par les courants. . . . . 617

§ 2. — Electro-dynamique. . . . 628

I. *Actions mutuelles des courants.* . 628

II. *Lois mathématiques des phénomènes électro-dynamiques.* . . 642

III. *Actions des aimants et de la terre sur les courants.* . . . 649

§ 3. — Théorie électro-dynamique du magnétisme. . . . . 656

I. *Aimantation des aimants aux solénoïdes.* . . . . . 656

II. *Rotation des courants par les aimants et réciproquement.* — Rotation des liquides. . . . . 662

§ 4. — Courants d'induction. . 670

I. *Induction par les courants et par les aimants.* — Induction par la terre. Machines magnéto-électriques. . . . . 670

II. *Lois et théorie des courants induits.* . . . . . 682

III. *Induction par les courants instantanés. Courants induits de différents ordres.* — Induction par la décharge. . . . . 699

IV. *Extra-courant. Appareil d'induction volta-électrique.* — Bobine de Ruhmkorff. Etincelle d'induction. Stratification de la lumière électrique. . . . . 711

V. *Magnétisme par mouvement.* — Disques. Masses de forme quelconque. Chaleur produite par induction.. . . . . 739

856 TABLE DES MATIÈRES.

§ 5. — Universalité du magnétisme. . . . . 753

I. *Diamagnétisme.* — Magnétisme des gaz. Polarité diamagnétique.. 753

II. *Magnétisme spécifique.* — Corps magnétiques. Actions mécaniques. Corps diamagnétiques. . . . . 773

III. *Magnétisme et diamagnétisme des cristaux.* . . . . . 785

§ 6. — Applications de l'électro-magnétisme. . . . . 788

I. *Moteurs électro-magnétiques.* . . 788

II. *Télégraphes électriques.* . . . . 798

III. *Horloges électriques.* . . . . . 832

IV. *Applications aux observations scientifiques.* — Chronoscopes. Appareils de météorologie. Régulateurs de la lumière électrique, etc. 838

FIN DE LA TABLE DES MATIÈRES.

Figure 99: Table of content for *DAGUIN, P.A., 1861. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 3, Edition 2.* (p. 355-358)

We also plotted the chapter on figure 77. [3] As we can see, the usage of features seems to be heavily influenced by the chapter, with chapters that use features with use several of them, while some chapters remain completely feature-less.

**Feature position in *MÜLLER, J. and C. S. M. Pouillet, 1906. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 10.*** We can do the same for a book with more false positives in order to get a complete picture of the potential of our tool. This time we use a custom threshold of 0.8.

[3] Because of a mismatch between the table of content and the pdf, our page numeration used in figures 78 to 98 are incremented by two compared to the pages used in this table of content.

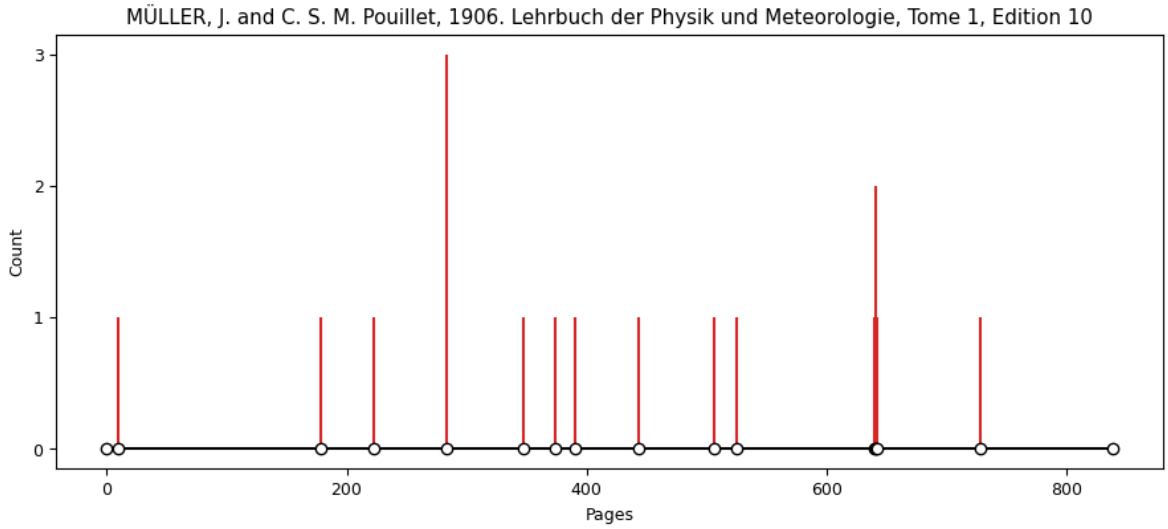


Figure 100: Predicted hands throughout MÜLLER, J. and C. S. M. Pouillet, 1906. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 10.

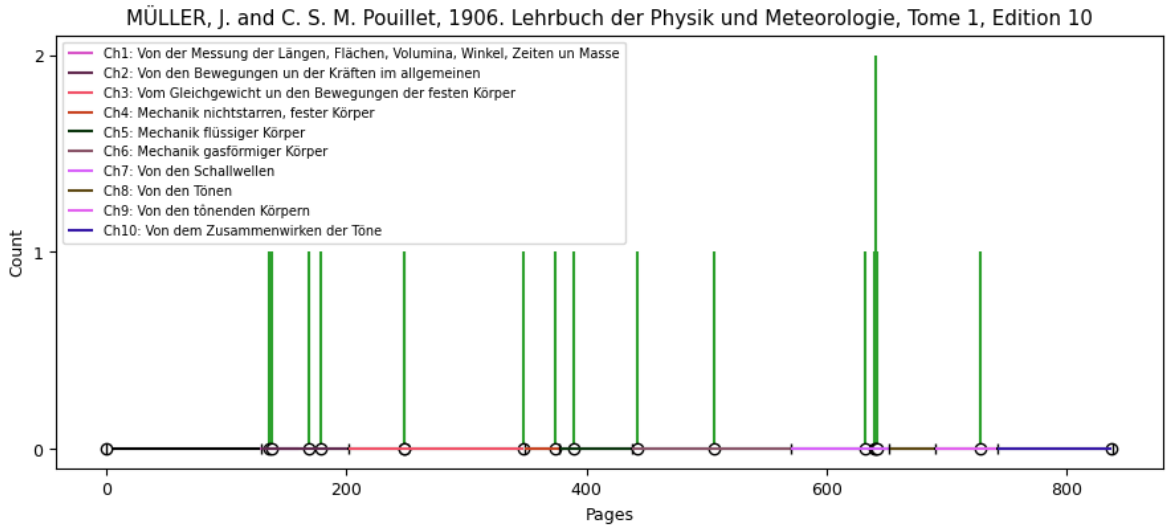


Figure 101: Manually found hands throughout MÜLLER, J. and C. S. M. Pouillet, 1906. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 10.


Figure 102: FP, p.10

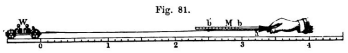


Figure 103: FN, p.136

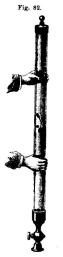

Figure 104: FN, p.138

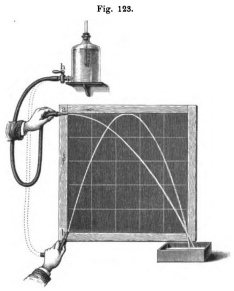


Figure 105: FN, p.169

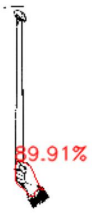


Figure 106: TP, p.179


Figure 107: FP, p.223

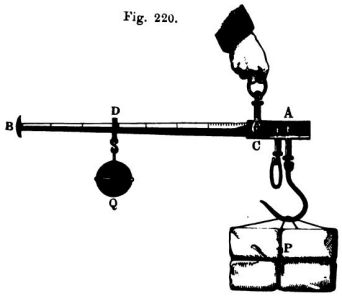


Figure 108: FN, p.248

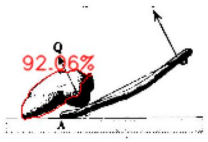


Figure 109: FP, p.283

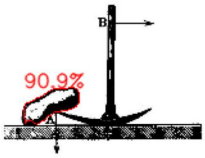


Figure 110: FP, 283

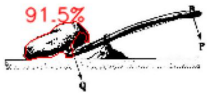


Figure 111: FP, p.283


Figure 112: TP, p.347

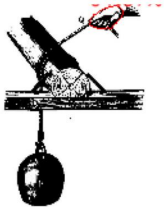

Figure 113: TP, p.374

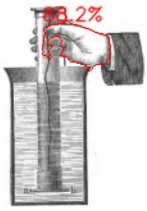


Figure 114: TP, p.390


Figure 115: TP, p.443


Figure 116: TP, p.506

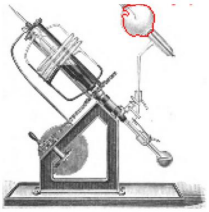

Figure 117: FP, p.525

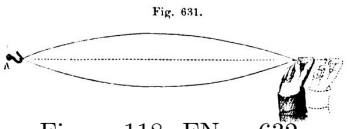


Figure 118: FN, p.632


Figure 119: TP, p.640

Figure 120: TP, p.641

Figure 121: TP, p.641

Figure 122: TP, p.642

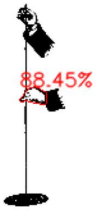


Figure 123: TP and FN, p.728

All these illustrations are taken from *MÜLLER, J. and C. S. M. Pouillet, 1906. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 10.*[4]

As for the previous book, we added as a caption if the detection is a true positive, false positive or false negative. As we can observe we have a few false positives with high confidence (figure 109 to 111 for example). Such false positives are often found on the part of illustrations that are circular with a large contour.

In figure 101, we plotted the chapters on the page axis. While one could argue there is still a trend in the usage of this particular feature for some chapters, such as chapter 7, especially pages talking about "Singing flames, analysing mirror, stroboscope method of investigation", it is less obvious than in figure 77.

The usage of this particular feature in this book seems to be a bit different from our previously studied case, as most hands seem to not be used for demonstrating an action but for displaying an object.

**Feature count in books** In order to further analyse the specific feature we searched for, i.e. hands we tried a more quantitative analysis by counting how many hands were detected by each book. This number had to be divided by the number of pages in that particular volume as some volumes have strong differences in terms of length as demonstrated in figure 124. This does mean that a book using more illustrations with hands but fewer illustrations overall would be penalised by the method, this is a bias to keep in mind.

[4]Because of the mismatch between the table of content and the pdf, our page numeration used in figures 102 to 123 are incremented by 24 compared to the pages used in the table of content.

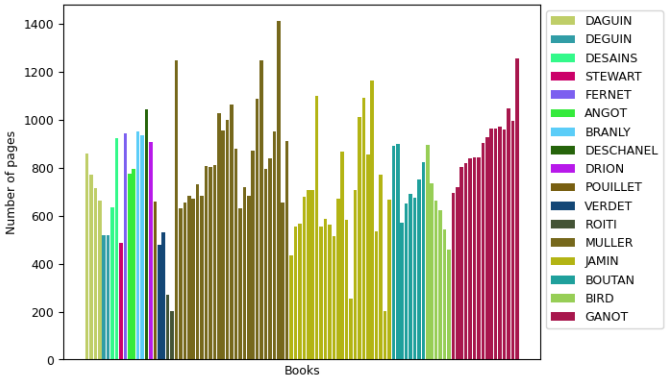


Figure 124: Number of pages per book

These metrics are also very dependent on the confidence level we require. As we can see in figure 125, 126 and 127. This is especially true for books from *BOUTAN*, we can see strong variations with different confidence levels, this was explored in paragraph 5, especially in figure 75 as we explained confidence distribution differences throughout authors.

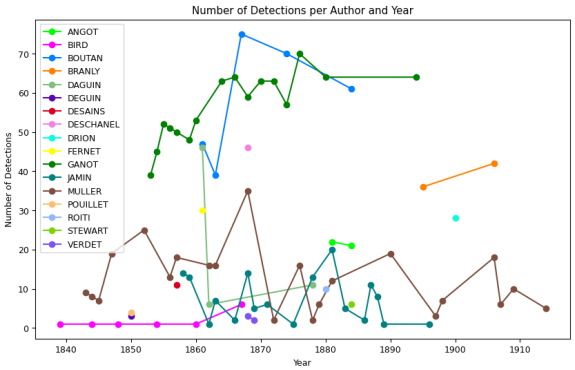


Figure 125: Hands detected for a confidence of 0.7

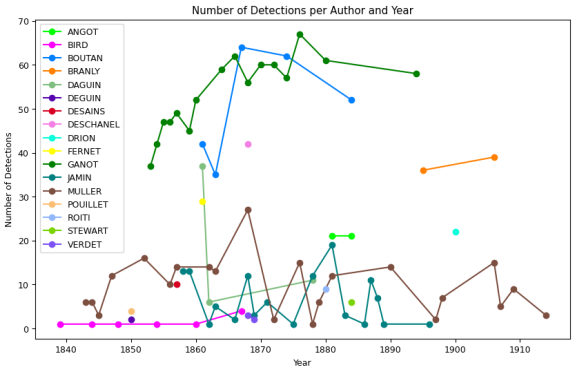


Figure 126: Hands detected for a confidence of 0.8

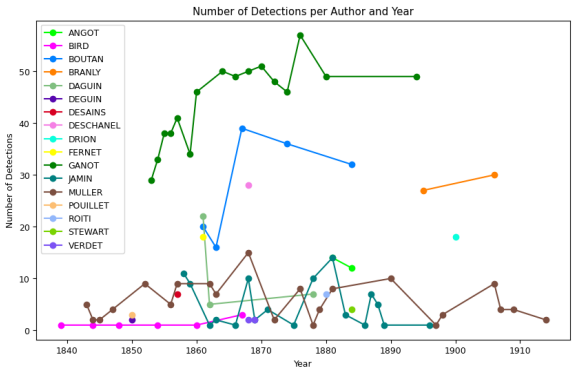


Figure 127: Hands detected for a confidence of 0.9

We will use a confidence level of 0.8 in the rest of these plots.

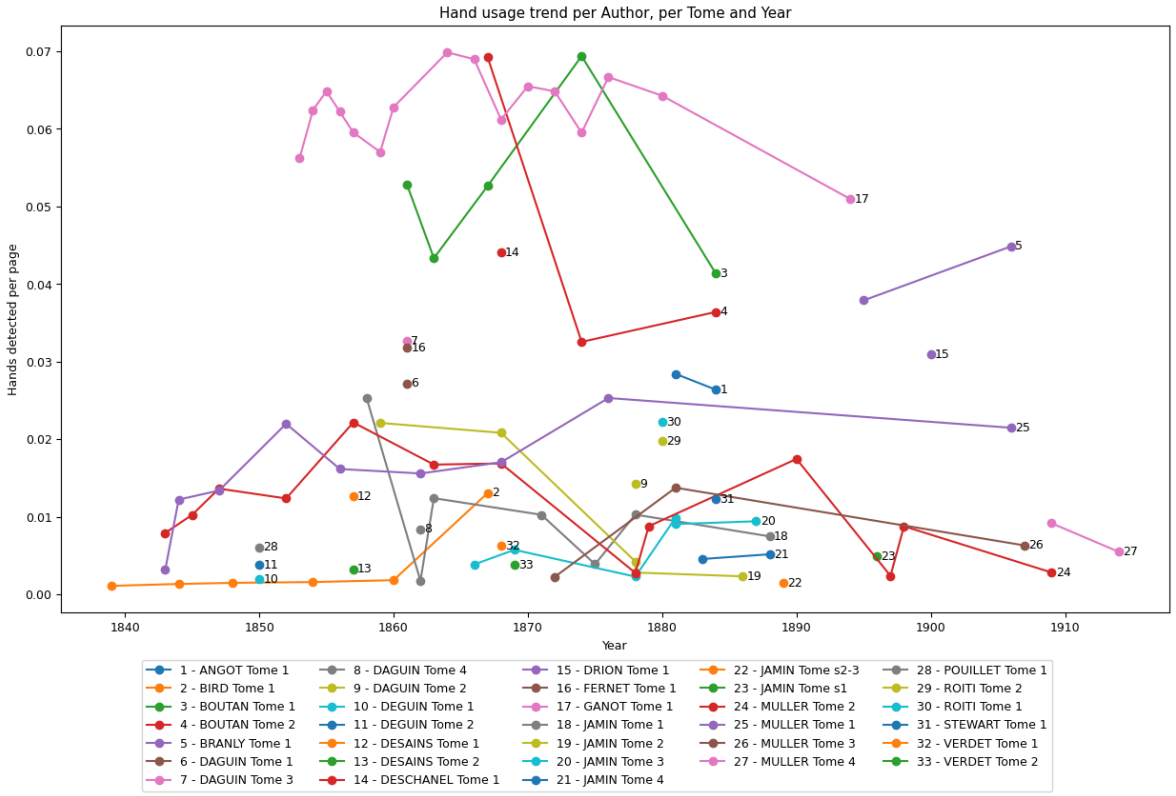


Figure 128: Number of hands detected per page for every volume of every author

As we discussed in section 5.1.1, chapters seem to influence the usage of particular features, for this reason, we decided to plot the number of hands by volumes in figure 128 so that the comparison would be more fair, as different editions from the same volume should have about the same chapters. We can first of all see a few outliers using a higher proportion of hands per page, such as volume 1 of author *GANOT*, volume 1 and 2 of author *BOUTAN*.

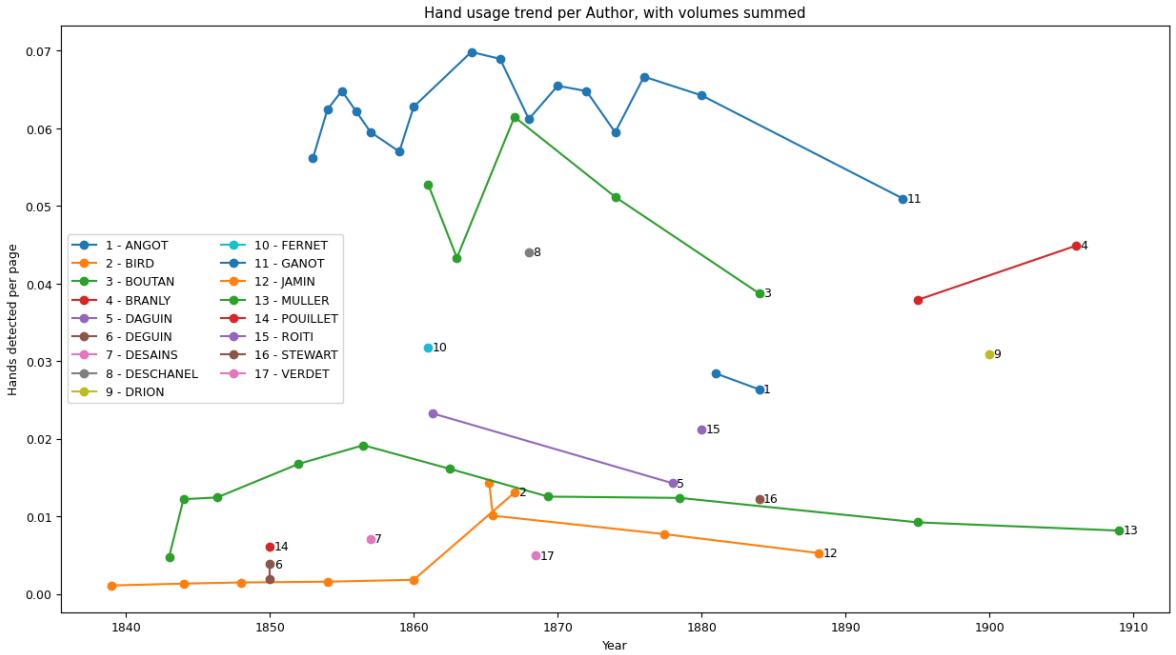


Figure 129: Number of hands detected per page across all editions

In figure 129, we also summed all volumes from a common edition together in order to more accurately have a sense of the author's overall usage of hands in illustrations as one volume could contain particular chapters that use hands as features, which would be highly ranked in figure 128 but that would only be because of the volume segmentation, which could be unfair as other authors may not segment chapters in the same way.
In figure 129, we still find that authors *GANOT* and *BOUTAN* use overall more hands in illustration. We can also observe that author *MULLER* seems to make less use of hands throughout the years, while other authors do not seem to follow any particular trend.

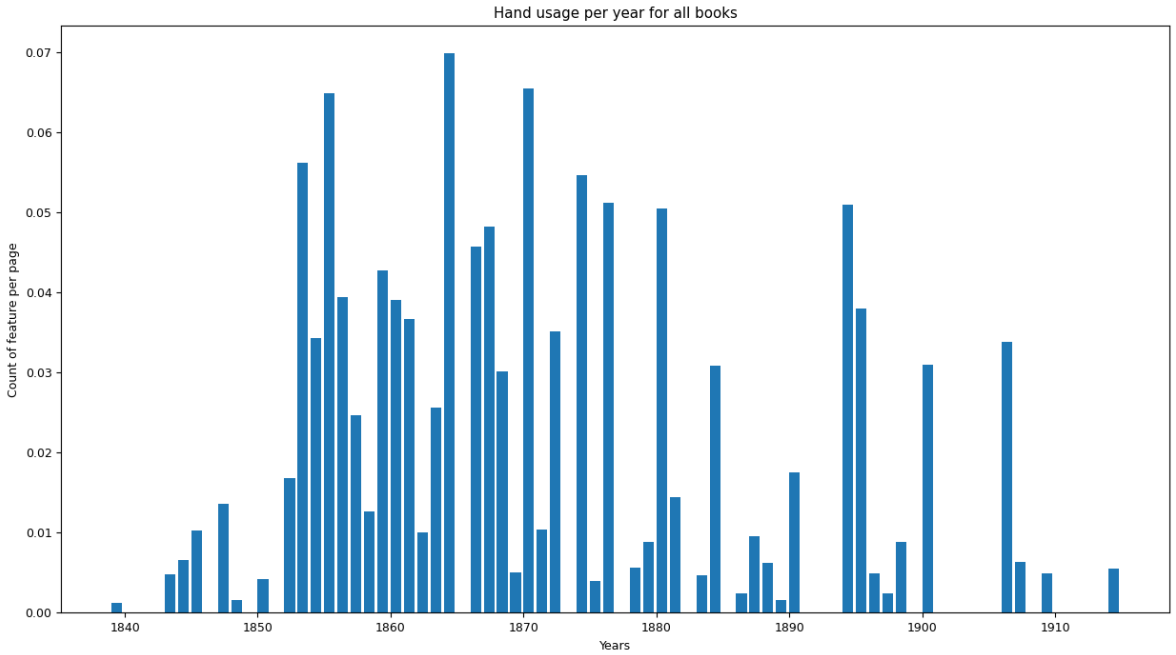


Figure 130: Bar plot of hand usage per year across all books per year

We also wished to maybe find some more trends in hand usage as features in illustrations, we also plotted in figure 130 the proportion of hands detected per page in all books from all authors per year or per decade in the case of figure 131. We do observe higher usage of hands from the 1850s to the 1900s, but this could very well be because of the publications of authors *GANOT* and *BOUTAN* across these periods that use overall more hands per page as many other authors.

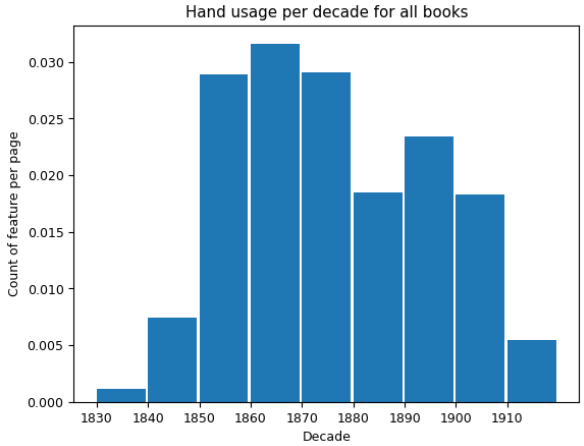


Figure 131: Bar plot of hand usage per year across all books per decade

### 5.1.2 Finding duplicates

By simply using PCA, we try to find duplicate images by selecting how similar images can be. This should in theory allow us to find images with a large number of similar features but that are not exact replicas. This method could in theory simply be used with all illustrations that we extract from the books (from paragraph 4.3). The computation of such a large similarity matrix for about 166295 images is computationally very expensive. This is the curse of dimensionality.
Selecting a subset of images, such as using only images with some feature allows us to reduce greatly that number while keeping images that we now are worthwhile comparing.

**Computing PCA:** In order to do a PCA on our illustration, we first need to select a number of components. These components are classified by explained variance, this means that the first component encapsulates more data about our illustrations than the second one and so on. With about 250 components, we explain about 90% of the variance in our illustrations.

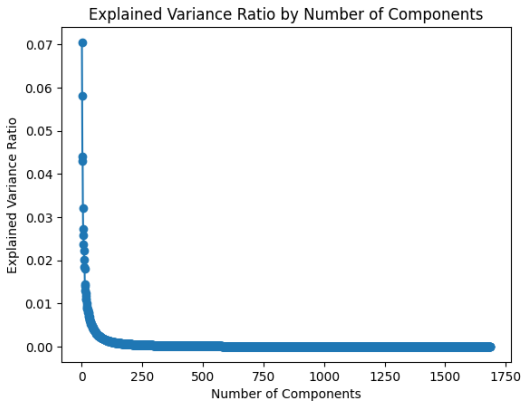


Figure 132: Explained variance ratio for each component, in decreasing order of importance

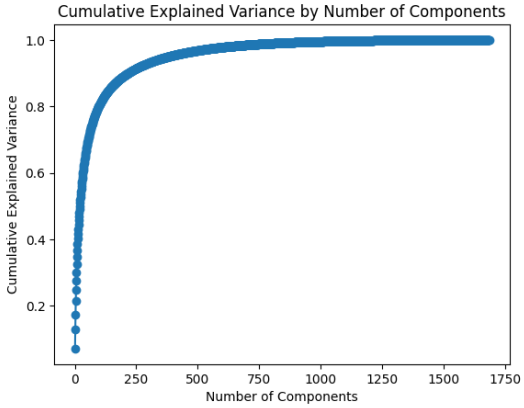


Figure 133: Cumulative explained variance by number of components

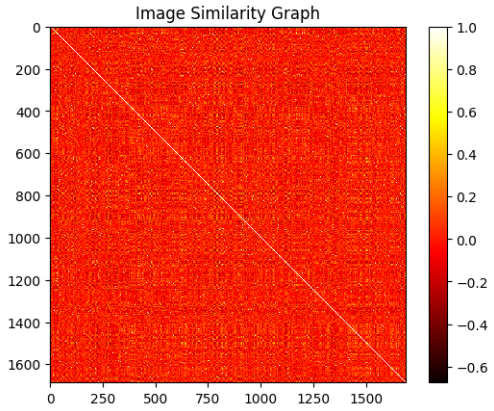


Figure 134: Similarity matrix between images using 250 components

This matrix represents image similarity between illustrations. As we can see, the diagonal has a similarity of 1 as the images compared to itself. Values with a large similarity have a value closer to 1.

| | BRANLY | GANOT | BOUTAN | MULLER | FERNET | ANGOT | JAMIN | DAGUIN | POUILLET | VERDET | DESCHANEL | DESAINS | ROITI | BIRD | STEWART |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **BRANLY** | 106 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **GANOT** | 1 | 14477 | 0 | 181 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 11 | 0 | 0 |
| **BOUTAN** | 0 | 0 | 644 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **MULLER** | 0 | 181 | 0 | 273 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **FERNET** | 2 | 1 | 0 | 0 | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **ANGOT** | 0 | 0 | 0 | 0 | 0 | 41 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 |
| **JAMIN** | 0 | 0 | 0 | 0 | 0 | 0 | 261 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **DAGUIN** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **POUILLET** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 |
| **VERDET** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| **DESCHANEL** | 0 | 1 | 0 | 0 | 0 | 4 | 0 | 0 | 1 | 0 | 30 | 0 | 0 | 0 | 0 |
| **DESAINS** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| **ROITI** | 0 | 11 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 |
| **BIRD** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| **STEWART** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

Table 1: Table of similar images between authors.

As we can see in table 1, the diagonal has the highest numbers of illustration similarity. This is because we use several books from the same authors from different editions, this means a lot of illustrations are reused when the book is updated. The number present in table 1 is heavily dependent

on the number of books per author we used in our dataset. We can see, a large number of books from authors *GANOT*, *JAMIN* and *MULLER* were used.

**Examples of duplicates:** We will exemplify the data we could extract from our PCA with two examples of illustrations reused with or without minor changes in figure 135 and 136.

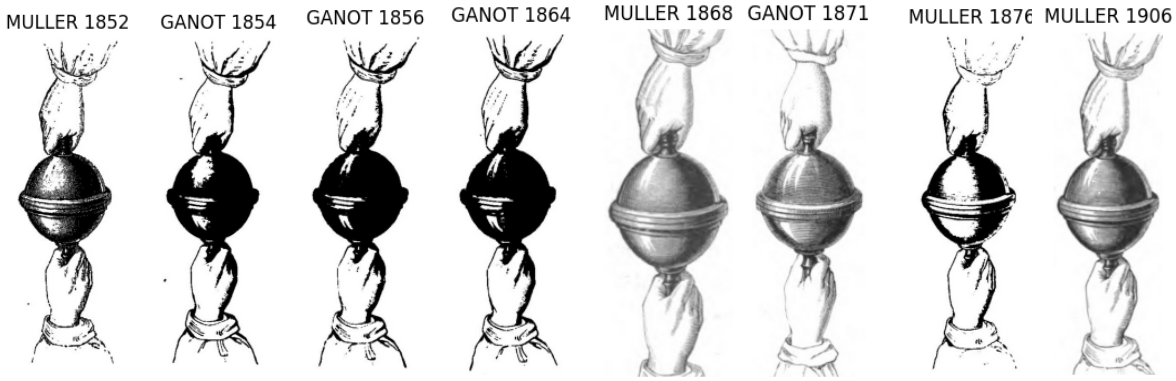


Figure 135: Usage of the same illustration across year from authors *GANOT* and *MULLER*

A lot of illustrations are shared between *GANOT* and *MULLER* as we see in table 1. As we can see in figure 135, not only the engraving plate are used by several authors but the illustrations also seem to go through some changes with time and have been redrawn.

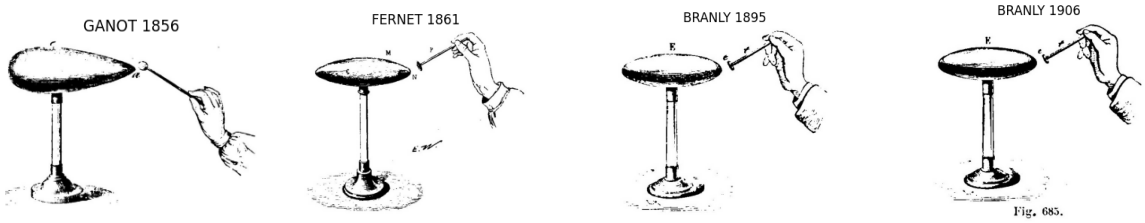


Figure 136: Usage of the same illustration across year from authors GANOT, FERNET and BRANLY

With more time, we could have used code from Ludovica Schaerf's thesis for clustering the visual signatures of artworks [Sch22] or even analyse the evolution in engraving plates[CAB+15].that seem to have been loaned between authors.

# 6 Discussion

**Limits in our performance comparison:** Some comparisons had to be made with slightly different parameters other than the one we compared for, this is simply because training models is very time-consuming and we were not able to train for every parameter. We had to rely upon the closest match.
Some questions about reproducibility were raised during the comparison, some experiences were reproduced with similar results but this was not always the case. This is also the reason models using bounding boxes were not used as we experienced technical difficulties that make the result time-consuming to reproduce.

Additionally, the choice of training set introduced bias as they were not picked at random from the final choice of books but from an initial number of books. There was an attempt to mitigate this by selecting a testing set from all books as to accurately portray our application performances.

With more time, we could have tried using transfer learning from the models made from "Hands Segmentation in PyTorch - A Plug and Play Model" [Cam21] to compare more accurately to models building using other datasets.

Performances due to the quality of book scans would also have been an interesting metric to compare, as this could tell us more about optimal scanning conditions analysis using deep learning applications.

Finally, using active learning to append our training dataset would have been an interesting approach to circumvent constraints due to the small number of illustrations containing features in our training set.

**Limits in our analysis:** As our models were trained with a subset of books, it is apparent that any representation of our features that do not fit the appearance of our features used in training may have trouble being detected.
Using only one confidence level for declaring predictions as positive in quantitative analysis introduce a bias for books that have a lot of false positive with high confidence.
Additionally, we did not spend much time tracing the origin of the illustration, as we just assumed they were somewhat tied to the author's work. An analysis of the relationship between illustrators and authors would have benefited this work.
We also regret not having the time to label and train models for other features listed in subsection 3.1, as the analysis of these features could have shed some more light on the history of physics book illustrations.

# 7 Conclusion

Out of necessity to accurately find features in illustrations in order to analyse them, this project ended up mainly comparing neural networks. This was however for the best as any models that would be used without checking for accuracy may have introduced a heavy bias that the user may not be aware of. In this report, we tried to aggregate all useful information that we have gathered throughout the semester, as this project could be a useful basis for continued research on this topic or on any topic using sketches or illustrations engraving. Even though no major trends were discovered in analysing hands as features in illustration, other features may tell us more about the history of physics books. The results are still satisfying in terms of hands as features in illustrations, we can still conclude that:

- The usage of hands seems to be localised to specific chapters.
- The usage of hands seems to be very dependent on the author.
- The function given to hands in illustrations seems to be dependent on the author/book.
- No major trends in time seem to be found in the usage of hands in illustrations.
- The reuse of engraving plates and the redrawing of other's author illustrations seem to have been common practice.

The use of image recognition for illustrations seems to be promising, offering far easier access to data than manual search without adding too much bias to the data.

# 8 Acknowledgement

A big thank you to everyone involved: Prof. Dr. Jérôme Baudry, Semion Sidorenko, Dr. Ion-Gabriel Mihailescu and Julien Metter for his previous work. While this project has been a very time-consuming endeavour, especially coming from micro-engineering, with little experience in applied machine learning using Python, it has been the most engaging project in my curriculum.

# 9 Annexe

## 9.1 Models performances

| | Architecture | Encoder | Encoder Weight | Loss | Dataset | Comment | iou | iou_iw | f1 | f1_iw | accuracy | accuracy_iw | recall | recall_iw | precision | precision_iw | fpr | fpr_iw |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | FPN | resnext101_32x8d | instagram | Diceloss | smaller_dataset | epoch100 | 0.389922 | 0.987787 | 0.561071 | 0.460548 | 0.999546 | 0.999547 | 0.411409 | 0.988986 | 0.881878 | 0.998311 | 0.000039 | 0.000040 |
| **1** | FPN | resnext101_32x8d | instagram | Diceloss | smaller_dataset | epoch100_1 | 0.106426 | 0.981978 | 0.192377 | 0.132618 | 0.999350 | 0.999350 | 0.109865 | 0.984580 | 0.772687 | 0.996191 | 0.000023 | 0.000023 |
| **2** | FPN | resnet50 | imagenet | Diceloss | hands_only | epoch100_lr_e-4 | 0.248199 | 0.982520 | 0.397691 | 0.351753 | 0.999303 | 0.999303 | 0.326602 | 0.986993 | 0.508337 | 0.994832 | 0.000223 | 0.000224 |
| **3** | FPN | resnet50 | imagenet | Diceloss | hands_only | epoch100_lr_e-5 | 0.213700 | 0.983910 | 0.352147 | 0.273234 | 0.999384 | 0.999384 | 0.237720 | 0.985807 | 0.678964 | 0.997471 | 0.000079 | 0.000081 |
| **4** | FPN | resnet50 | imagenet | Diceloss | hands_only | epoch100_lr_e-6 | 0.063774 | 0.982737 | 0.119902 | 0.081111 | 0.999310 | 0.999310 | 0.066720 | 0.984024 | 0.590892 | 0.998583 | 0.000033 | 0.000033 |
| **5** | FPN | resnet101 | imagenet | Diceloss | hands_only | epoch100_lr_e-6 | 0.098218 | 0.983218 | 0.178868 | 0.128358 | 0.999301 | 0.999301 | 0.108022 | 0.984353 | 0.519739 | 0.998301 | 0.000070 | 0.000073 |
| **6** | FPN | resnet152 | imagenet | Diceloss | hands_only | epoch100_lr_e-6 | 0.017414 | 0.983100 | 0.034231 | 0.021783 | 0.999303 | 0.999303 | 0.017532 | 0.983601 | 0.719977 | 0.999493 | 0.000005 | 0.000005 |
| **7** | FPN | resnext101_32x8d | instagram | Diceloss | hands_only | epoch100_lr_e-4 | 0.078741 | 0.971668 | 0.145987 | 0.202147 | 0.997759 | 0.997759 | 0.271870 | 0.986055 | 0.099784 | 0.985395 | 0.001729 | 0.001730 |
| **8** | FPN | resnext101_32x8d | instagram | Diceloss | hands_only | epoch100_lr_e-5 | 0.406946 | 0.983366 | 0.578481 | 0.493902 | 0.999538 | 0.999538 | 0.450036 | 0.988868 | 0.809529 | 0.994079 | 0.000075 | 0.000076 |
| **9** | FPN | resnext101_32x8d | instagram | Diceloss | hands_only | epoch100_lr_4.11E-06 | 0.401718 | 0.987203 | 0.573179 | 0.554868 | 0.999430 | 0.999430 | 0.543297 | 0.990361 | 0.606539 | 0.996138 | 0.000249 | 0.000250 |
| **10** | FPN | resnext101_32x8d | imagenet | Diceloss | hands_only | epoch100_lr_4.11E-06 | 0.227236 | 0.984894 | 0.370321 | 0.280078 | 0.999423 | 0.999423 | 0.240936 | 0.985970 | 0.799853 | 0.998594 | 0.000043 | 0.000043 |
| **11** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | hands_only | epoch100_lr_4.11E-06 | 0.541365 | 0.986990 | 0.702449 | 0.657138 | 0.999624 | 0.999624 | 0.630045 | 0.990844 | 0.793656 | 0.995833 | 0.000116 | 0.000118 |
| **12** | UnetPlusPlus | resnext101_32x16d | instagram | Diceloss | hands_only | epoch300_lr_4.11E-06 | 0.586317 | 0.987673 | 0.739218 | 0.687419 | 0.999674 | 0.999673 | 0.656739 | 0.991243 | 0.845391 | 0.996027 | 0.000085 | 0.000088 |
| **13** | UnetPlusPlus | resnext101_32x8d | imagenet | Diceloss | hands_only | epoch500_lr_4.11E-06 | 0.407243 | 0.982371 | 0.578781 | 0.564374 | 0.999431 | 0.999431 | 0.555161 | 0.990217 | 0.604500 | 0.990628 | 0.000256 | 0.000259 |
| **14** | UnetPlusPlus | resnext101_32x16d | ssl | Diceloss | hands_only | epoch500_lr_4.11E-06 | 0.377636 | 0.972678 | 0.548238 | 0.602836 | 0.999250 | 0.999250 | 0.645706 | 0.991805 | 0.476336 | 0.980560 | 0.000501 | 0.000503 |
| **15** | Unet | resnext101_32x8d | instagram | Diceloss | hands_only | epoch500_lr_4.11E-06 | 0.383762 | 0.977407 | 0.554665 | 0.576723 | 0.999330 | 0.999330 | 0.592430 | 0.990096 | 0.521427 | 0.986285 | 0.000383 | 0.000385 |
| **16** | Unet | resnext101_32x16d | instagram | Diceloss | hands_only | epoch500_lr_4.11E-06 | 0.516867 | 0.984864 | 0.681493 | 0.700347 | 0.999530 | 0.999530 | 0.713508 | 0.992849 | 0.652228 | 0.991616 | 0.000268 | 0.000272 |
| **17** | Unet | resnext101_32x8d | imagenet | Diceloss | hands_only | epoch500_lr_4.11E-06 | 0.222648 | 0.930391 | 0.364206 | 0.423668 | 0.998830 | 0.998830 | 0.475413 | 0.988763 | 0.295162 | 0.940488 | 0.000800 | 0.000804 |
| **18** | PSPNet | resnext101_32x8d | instagram | Diceloss | hands_only | epoch500_lr_4.11E-06 | 0.078817 | 0.881967 | 0.146117 | 0.236966 | 0.996667 | 0.996667 | 0.404729 | 0.989014 | 0.089151 | 0.889158 | 0.002916 | 0.002919 |
| **19** | UnetPlusPlus | resnext101_32x16d | instagram | Diceloss | hands_only | epoch500_lr_4.11E-06_CUSTOM_SPLIT-50-10-40 | 0.561123 | 0.989292 | 0.718871 | 0.717114 | 0.999605 | 0.999605 | 0.715947 | 0.992453 | 0.721819 | 0.996285 | 0.000195 | 0.000198 |
| **20** | UnetPlusPlus | resnext101_32x16d | instagram | Diceloss | hands_only | epoch500_lr_4.11E-06_CUSTOM_SPLIT-65-10-25 | 0.471752 | 0.987536 | 0.641075 | 0.656283 | 0.999474 | 0.999474 | 0.666828 | 0.992017 | 0.617237 | 0.994737 | 0.000292 | 0.000295 |
| **21** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | half_data | epoch500_lr_4.11 | 0.533906 | 0.989163 | 0.696139 | 0.627294 | 0.999638 | 0.999638 | 0.588494 | 0.990754 | 0.851980 | 0.997993 | 0.000072 | 0.000075 |
| **22** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | half_data | epoch2_lr_4.11 | 0.027010 | 0.798646 | 0.052600 | 0.119309 | 0.980400 | 0.980400 | 0.772203 | 0.994228 | 0.027227 | 0.801562 | 0.019453 | 0.019544 |
| **25** | UnetPlusPlus | resnext101_32x8d | instagram | TverskyLoss | half_data | epoch250_lr_1.20E-06 | 0.211196 | 0.835807 | 0.348739 | 0.505949 | 0.998096 | 0.998096 | 0.723334 | 0.992504 | 0.229755 | 0.839682 | 0.001710 | 0.001726 |
| **26** | UnetPlusPlus | resnext101_32x8d | instagram | TverskyLoss | half_data | epoch240_lr_0.0001 | 0.058820 | 0.959562 | 0.111104 | 0.139880 | 0.998094 | 0.998094 | 0.169074 | 0.984647 | 0.082736 | 0.973467 | 0.001322 | 0.001323 |
| **27** | UnetPlusPlus | resnext101_32x8d | instagram | TverskyLoss | half_data | epoch250_lr_0.00001 | 0.031647 | 0.528766 | 0.061353 | 0.136869 | 0.983555 | 0.983555 | 0.762779 | 0.993589 | 0.031962 | 0.530793 | 0.016290 | 0.016310 |
| **28** | UnetPlusPlus | resnext101_32x8d | instagram | TverskyLoss | half_data | epochlr_0.0001then0.0001 | 0.000662 | 0.000664 | 0.001324 | 0.003301 | 0.090617 | 0.090617 | 0.855222 | 0.997378 | 0.000662 | 0.000671 | 0.909922 | 0.909915 |
| **29** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | all_data | epoch_lr_4.11E-06 | 0.000000 | 0.983500 | 0.000000 | 0.000000 | 0.999295 | 0.999295 | 0.000000 | 0.983500 | NaN | 1.000000 | 0.000000 | 0.000000 |
| **31** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | smaller_dataset | epoch300_lr_4.11E-06 | 0.352240 | 0.986471 | 0.520972 | 0.580437 | 0.999186 | 0.999186 | 0.628243 | 0.990684 | 0.444991 | 0.995524 | 0.000552 | 0.000556 |
| **32** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | smaller_dataset+all_data | WITH2DATASET_epoch301_lr_4.11E-06 | 0.352240 | 0.986471 | 0.520972 | 0.580437 | 0.999186 | 0.999186 | 0.628243 | 0.990684 | 0.444991 | 0.995524 | 0.000552 | 0.000556 |
| **33** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | half_data | WEIRD_epoch300_lr_4.11E-06 | 0.352240 | 0.986471 | 0.520972 | 0.580437 | 0.999186 | 0.999186 | 0.628243 | 0.990684 | 0.444991 | 0.995524 | 0.000552 | 0.000556 |
| **36** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | half_split_test_val_distribution | batch32_size160_epoch200_lr_9.38E-05 | 0.202196 | 0.981011 | 0.336377 | 0.356426 | 0.998968 | 0.998968 | 0.371175 | 0.986873 | 0.307545 | 0.993625 | 0.000589 | 0.000593 |
| **37** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | half_split_test_val_distribution | batch_3_size480_epoch200_lr_9.38E-05 | 0.094134 | 0.975130 | 0.172070 | 0.217295 | 0.998214 | 0.998214 | 0.263458 | 0.985602 | 0.127755 | 0.988329 | 0.001268 | 0.001276 |
| **38** | UnetPlusPlus | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch32_size160_epoch4256_lr_1.09E-05 | 0.711307 | 0.989241 | 0.831302 | 0.835148 | 0.999760 | 0.999760 | 0.837731 | 0.996197 | 0.824971 | 0.992600 | 0.000125 | 0.000127 |
| **39** | UnetPlusPlus | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(33-3-3)-(80-1... | batch_25_size160_epoch3377_lr_1.09E-05 | 0.712563 | 0.989533 | 0.832160 | 0.818193 | 0.999770 | 0.999770 | 0.809139 | 0.995881 | 0.856529 | 0.993200 | 0.000096 | 0.000097 |
| **40** | UnetPlusPlus | resnext101_32x8d | instagram | dice_loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch32_size160_epoch4256_lr_1.09E-05 | 0.712425 | 0.991309 | 0.832065 | 0.778556 | 0.999788 | 0.999788 | 0.746549 | 0.994377 | 0.939708 | 0.996771 | 0.000034 | 0.000036 |
| **41** | UnetPlusPlus | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(10-3-3)-(80-1... | batch_25_size160_epoch2977_lr_1.09E-05 | 0.734209 | 0.993956 | 0.846737 | 0.800786 | 0.999803 | 0.999803 | 0.772827 | 0.995091 | 0.936278 | 0.998523 | 0.000037 | 0.000039 |
| **42** | UnetPlusPlus | resnext101_32x16d | instagram | dice_loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch16_size160_epoch4256_lr_1.09E-05 | 0.746705 | 0.992754 | 0.854987 | 0.811233 | 0.999812 | 0.999812 | 0.784469 | 0.995364 | 0.939436 | 0.997229 | 0.000036 | 0.000037 |
| **43** | UnetPlusPlus | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(70-3-3)-(80-1... | batch32_size160_epoch9000_lr_1.09E-05 | 0.574645 | 0.958234 | 0.729872 | 0.788717 | 0.999565 | 0.999565 | 0.833518 | 0.996249 | 0.649152 | 0.961712 | 0.000318 | 0.000320 |
| **44** | Unet | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch32_size160_epoch9000_lr_5.09E-05 | 0.688984 | 0.985966 | 0.815856 | 0.797672 | 0.999750 | 0.999750 | 0.785993 | 0.995221 | 0.848078 | 0.990375 | 0.000099 | 0.000102 |
| **45** | FPN | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch32_size160_epoch1321_lr_5.09E-05 | 0.574446 | 0.963299 | 0.729712 | 0.682296 | 0.999659 | 0.999659 | 0.653967 | 0.993930 | 0.825302 | 0.968893 | 0.000098 | 0.000100 |
| **46** | DeepLabV3Plus | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch32_size160_epoch1321_lr_5.09E-05 | 0.639846 | 0.991001 | 0.780373 | 0.755139 | 0.999707 | 0.999707 | 0.739203 | 0.994587 | 0.826400 | 0.995918 | 0.000109 | 0.000113 |
| **47** | PAN | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch32_size160_epoch1321_lr_5.09E-05 | 0.651868 | 0.983513 | 0.789249 | 0.751520 | 0.999726 | 0.999726 | 0.728310 | 0.994435 | 0.861318 | 0.988691 | 0.000083 | 0.000086 |
| **48** | MAnet | resnext101_32x8d | instagram | Jaccard_Loss | half_split_test_val_distribution(50-3-3)-(80-1... | batch32_size160_epoch1321_lr_5.09E-05 | 0.611763 | 0.981850 | 0.759123 | 0.721414 | 0.999688 | 0.999688 | 0.698290 | 0.993557 | 0.831567 | 0.987778 | 0.000100 | 0.000101 |
| **49** | UnetPlusPlus | resnext101_32x8d | instagram | JaccardLoss | half_split_test_val_distribution(50-3-3)-(80-1... | batch_16_size160_epoch1398_lr_5.09E-05 | 0.596636 | 0.985369 | 0.747366 | 0.754054 | 0.999639 | 0.999639 | 0.758579 | 0.995270 | 0.736480 | 0.989907 | 0.000191 | 0.000193 |

Table 2: Performances of models trained

## 9.2 Book used in this project

All illustrations found in this report were taken from the following books:

- French :
  - ANGOT, A., 1881. Traité de physique élémentaire. Librairie Hacette et Cie. Edition 1
  - ANGOT, A., 1884. Traité de physique élémentaire. Librairie Hacette et Cie. Edition 2
  - BRANLY, Edouard, 1895. Traité élémentaire de physique. Librairie chemin Poussielgue, Edition 1
  - BRANLY, Edouard, 1906. Traité élémentaire de physique. Librairie chemin Poussielgue, Edition 3
  - BOUTAN, A., D'ALMEIDA, J. CH., 1861. Cours élémentaire de physique. Dunod, Edition 1
  - BOUTAN, A., D'ALMEIDA, J. CH., 1863. Cours élémentaire de physique. Dunod, Edition 2
  - BOUTAN, A., D'ALMEIDA, J. CH., 1867. Cours élémentaire de physique. Dunod, Tome 1, Edition 3
  - BOUTAN, A., D'ALMEIDA, J. CH., 1867. Cours élémentaire de physique. Dunod, Tome 2, Edition 3
  - BOUTAN, A., D'ALMEIDA, J. CH., 1874. Cours élémentaire de physique. Dunod, Tome 1, Edition 4
  - BOUTAN, A., D'ALMEIDA, J. CH., 1874. Cours élémentaire de physique. Dunod, Tome 2, Edition 4
  - BOUTAN, A., D'ALMEIDA, J. CH., 1884. Cours élémentaire de physique. Dunod, Tome 1, Edition 5
  - BOUTAN, A., D'ALMEIDA, J. CH., 1884. Cours élémentaire de physique. Dunod, Tome 2, Edition 5
  - DAGUIN, P.A., 1861. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 3, Edition 2
  - DAGUIN, P.A., 1878. Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 2, Edition 4
  - DAGUIN, P.A., 1862.Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 4, Edition 2
  - DAGUIN, P.A., 1861.Traité élémentaire de physique théorique et expérimentale. Edouard privat. Tome 1, Edition 2
  - DEGUIN, M, 1850. Cours élémentaire de physique. Tome 1, Edition 7
  - DEGUIN, M, 1850. Cours élémentaire de physique. Tome 2, Edition 9
  - DESAINS, PAUL, 1857. Leçons de physique. Dezobry, E. Magdeleine et Cie. Tome 1, Edition 1
  - DESAINS, PAUL, 1857. Leçons de physique. Dezobry, E. Magdeleine et Cie. Tome 2, Edition 1
  - DESCHANEL, A, 1868. Traité élémenatire de physique. Libraire L. Hachette et Cie.
  - DRION, Charles Alexandre, FERNET, Emile. 1861. Traité de physique élémentaire
  - DRION, Charles Alexandre, FERNET, Emile, FAIBRE-DUPAIGRE, Jules, 1900. Traité de physique élémentaire. Masson et Cie, Edition 13
  - GANOT, A., 1853. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 2
  - GANOT, A., 1854. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 3

– GANOT, A., 1855. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 5

– GANOT, A., 1856. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 6

– GANOT, A., 1857. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 7

– GANOT, A., 1859. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 8

– GANOT, A., 1860. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 9

– GANOT, A., 1864. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 11

– GANOT, A., 1866. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur. Edition 12

– GANOT, A., 1868. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 13

– GANOT, A., 1870. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 14

– GANOT, A., 1871. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 15

– GANOT, A., 1874. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 16

– GANOT, A., 1876. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur. Edition 17

– GANOT, A., 1880. Traité élémentaire de physique expérimentale et appliquée. Chez l'auteur-éditeur. Edition 18

– GANOT, A., 1894. Traité élémentaire de physique expérimentale et appliquée. Librairie Hacette et Cie. Edition 21

– POUILLET, M, 1850. Notions générales de physique et météorologie. Auguste Pagny, Edition 3

– ROITI, Antonio, 1880 Elementi di fisics. Tome 1

– ROITI, Antonio, 1880 Elementi di fisics. Tome 2

– VERDET, Emile, 1868. Cours de physique. Victor Masson et Fils, Tome 1

– VERDET, Emile, 1869. Cours de physique. Victor Masson et Fils, Tome 2

– JAMIN, Jules, 1878. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 3

– JAMIN, Jules, 1858. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 1

– JAMIN, Jules, 1862. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Edition 2, Tome 1

– JAMIN, Jules, 1866. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 3, Partie 2

– JAMIN, Jules, 1859. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 2

– JAMIN, Jules, 1866. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 3

– JAMIN, Jules, 1878. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Edition 3, Tome 1

  - JAMIN, Jules, 1878. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Edition 3, Tome 2
  - JAMIN, Jules, 1878. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Edition 3, Tome 2
  - JAMIN, Jules, 1881. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Edition 3, Tome 3
  - JAMIN, Jules, 1858. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 1, Edition 1
  - JAMIN, Jules, 1859. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 2, Edition 1
  - JAMIN, Jules, 1863. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 1, Edition 2
  - JAMIN, Jules, 1866. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 3, Edition 1
  - JAMIN, Jules, 1868. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 2, Edition 2
  - JAMIN, Jules, 1869. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 3, Edition 2
  - JAMIN, Jules, 1871. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 1, Edition 3
  - JAMIN, Jules, 1875. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 1, Edition 3, Supplément 1
  - JAMIN, Jules, 1878. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 2, Edition 3
  - JAMIN, Jules, 1881. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 3, Edition 3
  - JAMIN, Jules, 1883. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 4, Edition 4
  - JAMIN, Jules, 1886. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 2, Edition 4
  - JAMIN, Jules, 1887. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 3, Edition 4
  - JAMIN, Jules, 1888. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 1, Edition 4
  - JAMIN, Jules, 1888. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Tome 4, Edition 4
  - JAMIN, Jules, 1896. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Edition 4, Supplément 1
  - JAMIN, Jules, 1889. Cours de physique de l'École Polytechnique. Mallet-Bachelier, Edition 4, Supplément 2-3

- German
  - MÜLLER, J. and C. S. M. Pouillet, 1906. Lehrbuch der Physik und Meteorologie, Tome 1
  - MÜLLER, J. and C. S. M. Pouillet, 1852. Lehrbuch der Physik und Meteorologie, Tome 2
  - MÜLLER, J. and C. S. M. Pouillet, 1843. Lehrbuch der Physik und Meteorologie, Tome 1
  - MÜLLER, J. and C. S. M. Pouillet, 1881. Lehrbuch der Physik und Meteorologie, Tome 3, Edition 10
  - MÜLLER, J. and C. S. M. Pouillet, 1843. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 1

– MÜLLER, J. and C. S. M. Pouillet, 1844. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 2
– MÜLLER, J. and C. S. M. Pouillet, 1845. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 2
– MÜLLER, J. and C. S. M. Pouillet, 1847. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 3
– MÜLLER, J. and C. S. M. Pouillet, 1847. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 3
– MÜLLER, J. and C. S. M. Pouillet, 1852. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 4
– MÜLLER, J. and C. S. M. Pouillet, 1852. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 4
– MÜLLER, J. and C. S. M. Pouillet, 1856. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 5
– MÜLLER, J. and C. S. M. Pouillet, 1857. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 5
– MÜLLER, J. and C. S. M. Pouillet, 1862. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 6
– MÜLLER, J. and C. S. M. Pouillet, 1863. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 6
– MÜLLER, J. and C. S. M. Pouillet, 1868. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 7
– MÜLLER, J. and C. S. M. Pouillet, 1868. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 7
– MÜLLER, J. and C. S. M. Pouillet, 1872. Lehrbuch der Physik und Meteorologie, Tome 3, Edition 7
– MÜLLER, J. and C. S. M. Pouillet, 1876. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 8
– MÜLLER, J. and C. S. M. Pouillet, 1878. Lehrbuch der Physik und Meteorologie, Tome 2, Partie 1, Edition 8
– MÜLLER, J. and C. S. M. Pouillet, 1879. Lehrbuch der Physik und Meteorologie, Tome 2, Partie 2, Edition 8
– MÜLLER, J. and C. S. M. Pouillet, 1881. Lehrbuch der Physik und Meteorologie, Tome 3, Edition 8
– MÜLLER, J. and C. S. M. Pouillet, 1890. Lehrbuch der Physik und Meteorologie, Tome 2, Partie 1, Edition 9
– MÜLLER, J. and C. S. M. Pouillet, 1897. Lehrbuch der Physik und Meteorologie, Tome 2, Partie 1, Edition 9
– MÜLLER, J. and C. S. M. Pouillet, 1898. Lehrbuch der Physik und Meteorologie, Tome 2, Partie 2, Edition 9
– MÜLLER, J. and C. S. M. Pouillet, 1906. Lehrbuch der Physik und Meteorologie, Tome 1, Edition 10
– MÜLLER, J. and C. S. M. Pouillet, 1907. Lehrbuch der Physik und Meteorologie, Tome 3, Edition 10
– MÜLLER, J. and C. S. M. Pouillet, 1909. Lehrbuch der Physik und Meteorologie, Tome 2, Edition 10
– MÜLLER, J. and C. S. M. Pouillet, 1909. Lehrbuch der Physik und Meteorologie, Tome 4, Partie 1, Edition 10
– MÜLLER, J. and C. S. M. Pouillet, 1914. Lehrbuch der Physik und Meteorologie, Tome 4, Partie 2, Edition 10

- English
  - BIRD, Goldin, 1867. Elements of Natural Philosophy. John Churchill, Edition 1
  - BIRD, Goldin, 1867. Elements of Natural Philosophy. John Churchill, Edition 2
  - BIRD, Goldin, 1867. Elements of Natural Philosophy. John Churchill, Edition 3
  - BIRD, Goldin, 1867. Elements of Natural Philosophy. John Churchill, Edition 4
  - BIRD, Goldin, 1867. Elements of Natural Philosophy. John Churchill, Edition 5
  - BIRD, Goldin, 1867. Elements of Natural Philosophy. John Churchill, Edition 6
  - STEWART, Balfour, 1884. Lessons in elementary physics. Macmillan and Co.

# References


[BLCY15] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[Bro20] Jason Brownlee. How to control the stability of training neural networks with the batch size, Aug 2020.

[CAB+15] Joon Son Chung, Relja Arandjelović, Giles Bergel, Alexandra Franklin, and Andrew Zisserman. Re-presentations of art collections. In *Computer Vision - ECCV 2014 Workshops*, pages 85–100. Springer International Publishing, 2015.

[Cam21] Guglielmo Camporese. Hands segmentation is all you need. *https://github.com/guglielmocamporese*, 2021.

[CZP+18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[Deva] Google Developers. Classification: Precision and recall - machine learning - google for developers. Accessed on Mai 28, 2023.

[Devb] Google Developers. Classification: Roc curve and auc - machine learning - google for developers. Accessed on Mai 28, 2023.

[FRR11] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011.

[FWLW20] Tongle Fan, Guanglei Wang, Yan Li, and Hongrui Wang. Ma-net: A multi-scale attention network for liver and tumor segmentation. *IEEE Access*, 8:179656–179665, 2020.

[Hui] Jonathan Hui. Understanding Feature Pyramid Networks for object detection (FPN) — jonathan-hui.medium.com. https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c. [Accessed 06-Jun-2023].

[Iak19] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019.

[KHGD17] Alexander Kirillov, Kaiming He, Ross Girshick, and Piotr Dollár. A unified architecture for instance and semantic segmentation, 2017.

[Kor] Joos Korstanje. The f1 score — towards data science. Accessed on Mai 28, 2023.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[LYR15] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 287–295, 2015.

[Met22] Julien Metter. Bachelor project report : Distant seeing applying machine vision algorithms to historical scientific images, 2022.

[MLW19] Zheng Ma, Ming Li, and Yuguang Wang. Pan: Path integral based convolution for deep graph neural networks. 2019.

[MOK] Mohammed El Amine MOKHTARI. The Difference Between Dice and Dice Loss - PYCAD — pycad.co. https://pycad.co/the-difference-between-dice-and-dice-loss/. [Accessed 06-Jun-2023].

[Pea96] Karl Pearson. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318, 1896.

[Pyta] Pytorch. fasterrcnn_resnet50_fpn_v2 — torchvision main documentation. Accessed on Mai 25, 2023.

[Pytb] Pytorch. fcos_resnet50_fpn — torchvision main documentation. Accessed on Mai 25, 2023.

[Pytc] Pytorch. retinanet_resnet50_fpn — torchvision main documentation. Accessed on Mai 25, 2023.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[RW16] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Fatih Porikli, Sandra Skaff, Alireza Entezari, Jianyuan Min, Daisuke Iwai, Amela Sadagic, Carlos Scheidegger, and Tobias Isenberg, editors, *Advances in Visual Computing*, pages 234–244, Cham, 2016. Springer International Publishing.

[Sch22] Ludovica Schaerf. *Semi-supervised Clustering of Visual Signatures of Artworks*. Phd thesis, EPFL, Lausanne, CH, July 2022.

[SEG17] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks, 2017.

[SLV+17] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer International Publishing, 2017.

[UB18] Aisha Urooj and Ali Borji. Analysis of hand segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4710–4719, 2018.

[UDT] Universaldatatool/universal-data-tool: Collaborate & label any type of data, images, text, or documents, in an easy web interface or desktop app. Accessed on Mai 26, 2023.

[VSW+22] Taha ValizadehAslani, Yiwen Shi, Jing Wang, Ping Ren, Yi Zhang, Meng Hu, Liang Zhao, and Hualou Liang. Two-stage fine-tuning: A novel strategy for learning class-imbalanced data. *arXiv preprint arXiv:2207.10858*, 2022.

[Wig19] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[Wik23a] Wikipedia. F-score — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=F-score&oldid=1148225663, 2023. [Online; accessed 31-May-2023].

[Wik23b] Wikipedia. False positive rate — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=False%20positive%20rate&oldid=1155130470, 2023. [Online; accessed 05-June-2023].

[XGD+16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2016.

[YJC+19] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification, 2019.

[ZSTL18] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation, 2018.