Kunisch Recognition through Multi-Label Classification Algorithms

Prof. Benjamín Bustos

Prof. Iván Sipirán

Matías Vergara

1. Introduction

Ornamente Geometrischer Vasen: Ein Kompendium[1] is a collection of patterns present on archaeological objects, each of which is labeled with a set of labels.

This work seeks to apply multi-label classification algorithms on them, in order to develop a tool that serves for the labeling of new entries in the future.

[1]Norbert Kunisch, 1998

2. Data Extraction

In order to conduct such experiments, it was first necessary to get a usable version of the labels and a descriptor for each pattern.

2. Data Extraction

In order to conduct such experiments, it was first necessary to get a usable version of the labels and a descriptor for each pattern.

For the labels:

- We decided to use the English version of labels, due to some good properties of the language (such as gender neutrality).
- Labels were extracted manually.

7b Opposed diagonals, separated by solid triangles, horizontal panel
Traits obliques affrontés, séparés par des triangles noirs, panneau horizontal
Triangoli di vernice, fra tratti intrecciati diagonalemente, campo orizzontale
Εναλλασσόμενες λοξές γραμμές, ενδιάμεσα μελαμβαφή τρίγωνα, οριζόντια ζώνη

7c Gegenständige Diagonalen, dazwischen Firnisdreiecke, Senkrechtfeld
Opposed diagonals, separated by solid triangles, vertical panel
Traits obliques affrontés, séparées par des triangles noirs , panneau vertical
Triangoli di vernice, fra tratti intrecciati diagonalemente, campo verticale
Εναλλασσόμενες λοξές γραμμές, ενδιάμεσα μελαμβαφή τρίγωνα, κάθετη ζώνη

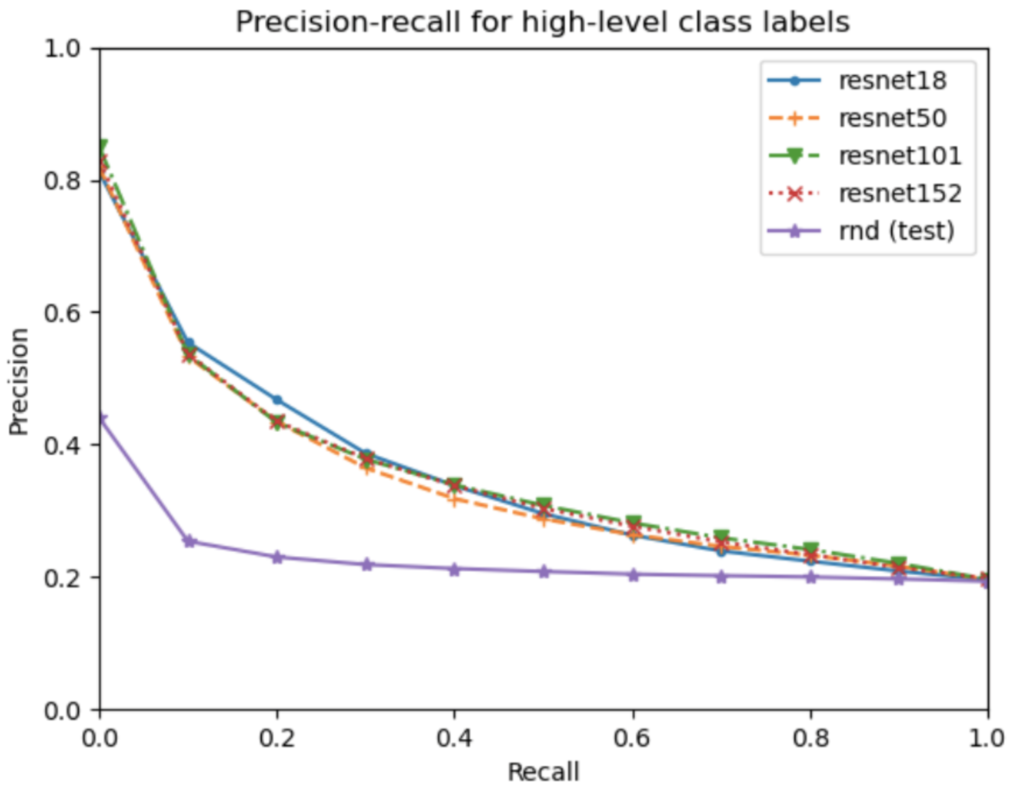
2. Data Extraction

In order to conduct such experiments, it was first necessary to get a usable version of the labels and a descriptor for each pattern.

For the descriptors:

- We conducted experiments testing ResNet architectures and a Random Neural Network in the task of classifying patterns in their high-level classes (chapter of the book).

- We decided to use ResNet18 features as descriptors for the patterns, and we also took the ones from ResNet50 in order to have an option to compare with.



3. Data Exploration

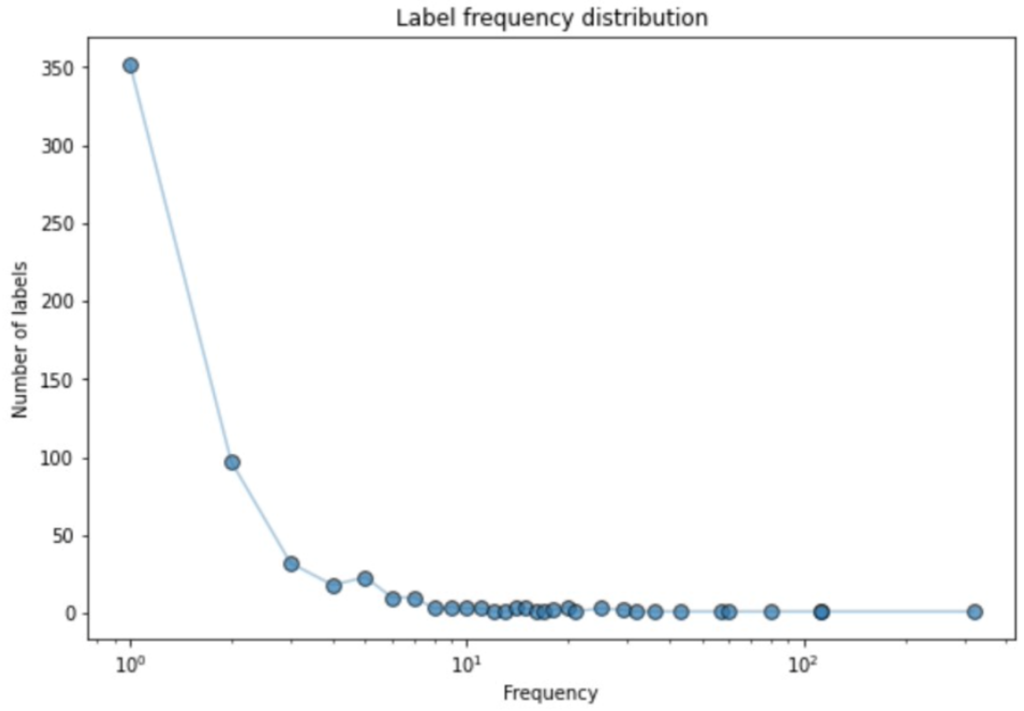
Once we had the data, we did an exploratory analysis to determine its principal properties. We discovered two main issues.

3. Data Exploration

Once we had the data, we did an exploratory analysis to determine its principal properties. We discovered two main issues.

Too many labels with low number of events in the data:

- For low numbers of events there are many labels, while for high occurrences there are very few labels.
- As example, there are 352 labels with a single event in the data, and only 3 with more than 100.
- Zipf-like distribution?



3. Data Exploration

Once we had the data, we did an exploratory analysis to determine its principal properties. We discovered two main issues.

Consequently, an extremely low *label density*[2]:

- Relation between the number of samples and labels related to each of them (which could be seen as how multi-label is the data).
- In our case, this measure takes a value of 0.005, extremely low.
- This is not surprising, however, since label cardinality (average number of labels per example) is very low relative to the total of labels (4 vs 586).

$$LD(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i|}{|L|}$$

[2]Tsoumakas G., Katakis I., V. I., "A review of multi-label classification methods," International Journal of Data Warehousing and Mining, vol. 3, pp. 1–13, 2007.

4. Proposed Methods

We decided to address the problem from two of the approaches most recommended by the literature.

4. Proposed Methods

We decided to address the problem from two of the approaches most recommended by the literature.

Through Problem Transformation Methods:

- Binary Relevance
- Power Labelset
- Classifier Chain
- Distinct Random k-Labelsets (RakELD)

4. Proposed Methods

We decided to address the problem from two of the approaches most recommended by the literature.

Through Problem Transformation Methods:

- Binary Relevance
- Power Labelset
- Classifier Chain
- Distinct Random k-Labelsets (RakELD)

Through Algorithm Adaptation Methods:

- Multilabel k-Nearest Neighbors (MLkNN)
- Twin Multi-label Support Vector Machines (MLTSVM)
- Binary Relevance K-Nearest Neighbors (BRkNN)

4. Proposed Methods

We decided to address the problem from two of the approaches most recommended by the literature.

Through Problem Transformation Methods:

- Binary Relevance
- Power Labelset
- Classifier Chain
- Distinct Random k-Labelsets (RakELD)

Through Algorithm Adaptation Methods:

- Multilabel k-Nearest Neighbors (MLkNN)
- Twin Multi-label Support Vector Machines (MLTSVM)
- Binary Relevance K-Nearest Neighbors (BRkNN)

For Problem Transformation, we tried with SVM and Logistic Regression as base classifiers.

For Algorithm Adaptation, we did a grid search to find best hyperparameters.

Experiments were conducted through the BSD-licensed **scikit-multilearn** library.

5. Metrics

Finally, it is important to pay attention to which metrics will we use to compare results, since the evaluation of a multi-label classification algorithm includes an additional notion of being partially correct.

5. Metrics

Finally, it is important to pay attention to which metrics will we use to compare results, since the evaluation of a multi-label classification algorithm includes an additional notion of being partially correct.

We decided to use 3 metrics:

- **Exact Match Ratio**
- **Hamming Loss**
- **Hamming Score** (also knew as label-based accuracy).

$$\text{Hamming-Loss} = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{Y_i \Delta Z_i}{M}\right|$$

$$\text{Hamming-Score} = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{Y_i \cap Z_i}{Y_i \cup Z_i}\right|$$

6. Pruning Strategy

To treat the issues related to the problem’s *label density,* we adopted a **problem pruning approach.**

6. Pruning Strategy

To treat the issues related to the problem's *label density,* we adopted a **problem pruning approach.**

We prune the labels based on their number of events in the data set. We then apply and compare methods at different levels of pruning.

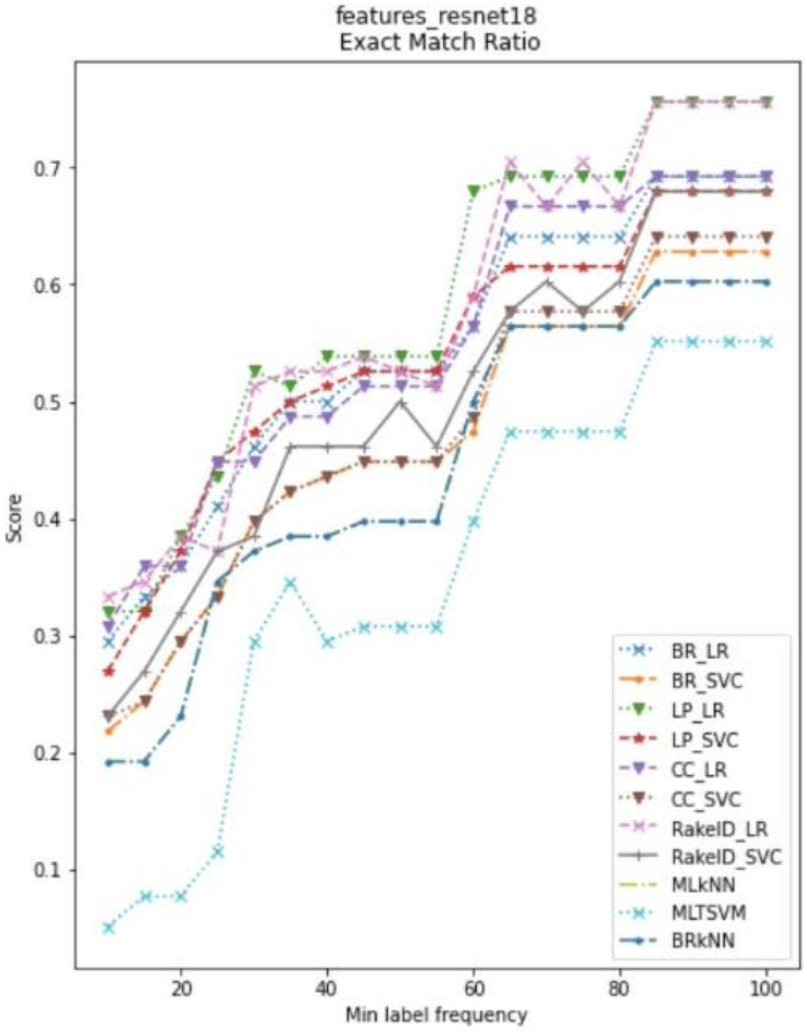
6. Pruning Strategy

To treat the issues related to the problem’s *label density,* we adopted a **problem pruning approach.**

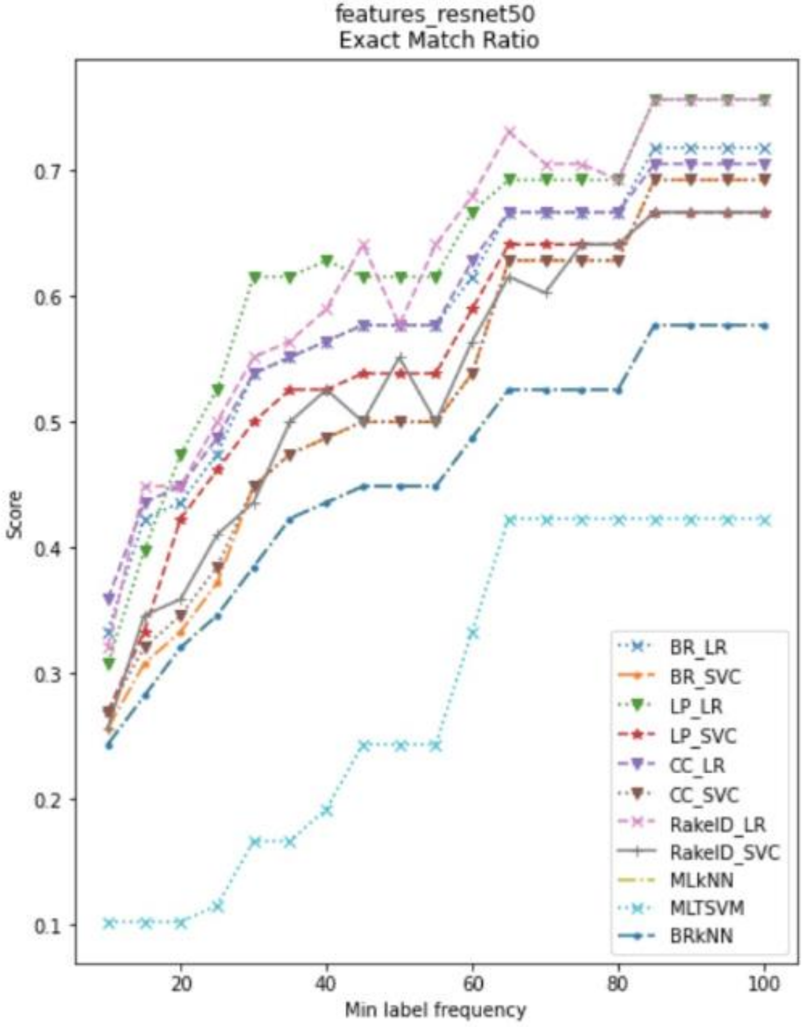
We prune the labels based on their number of events in the data set. We then apply and compare methods at different levels of pruning.

The more we prune, the easier the problem… But it also becomes less interesting.

7. Results

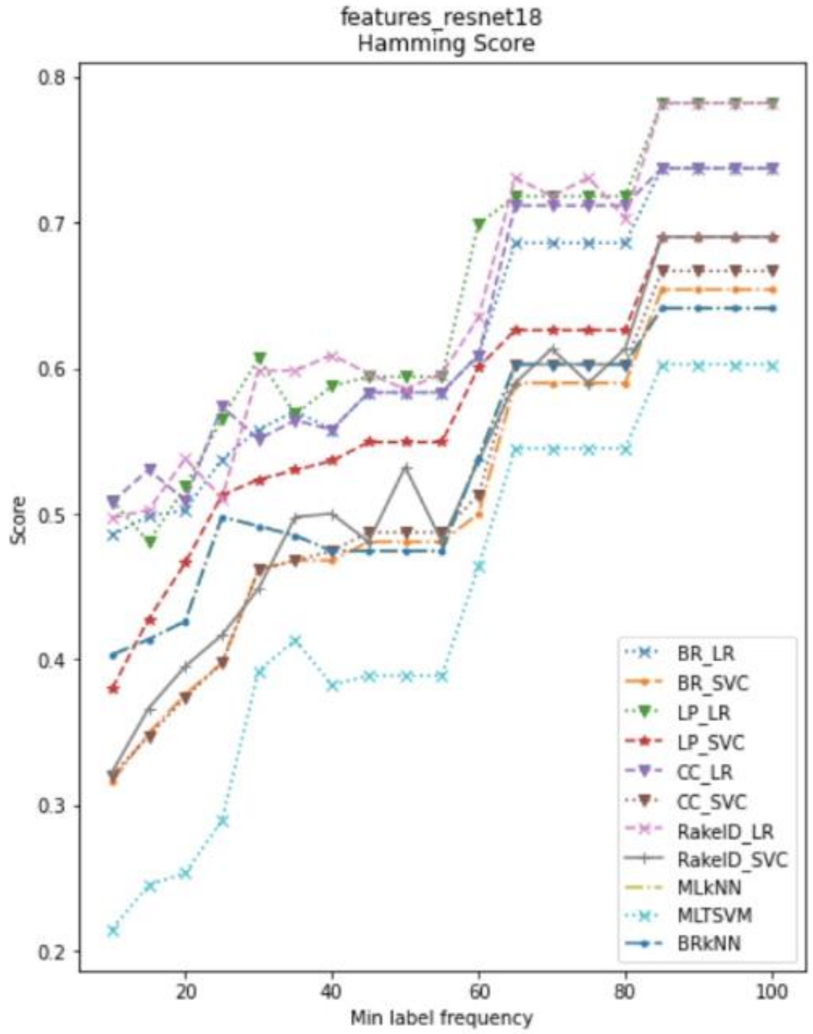


(a) Features from ResNet18

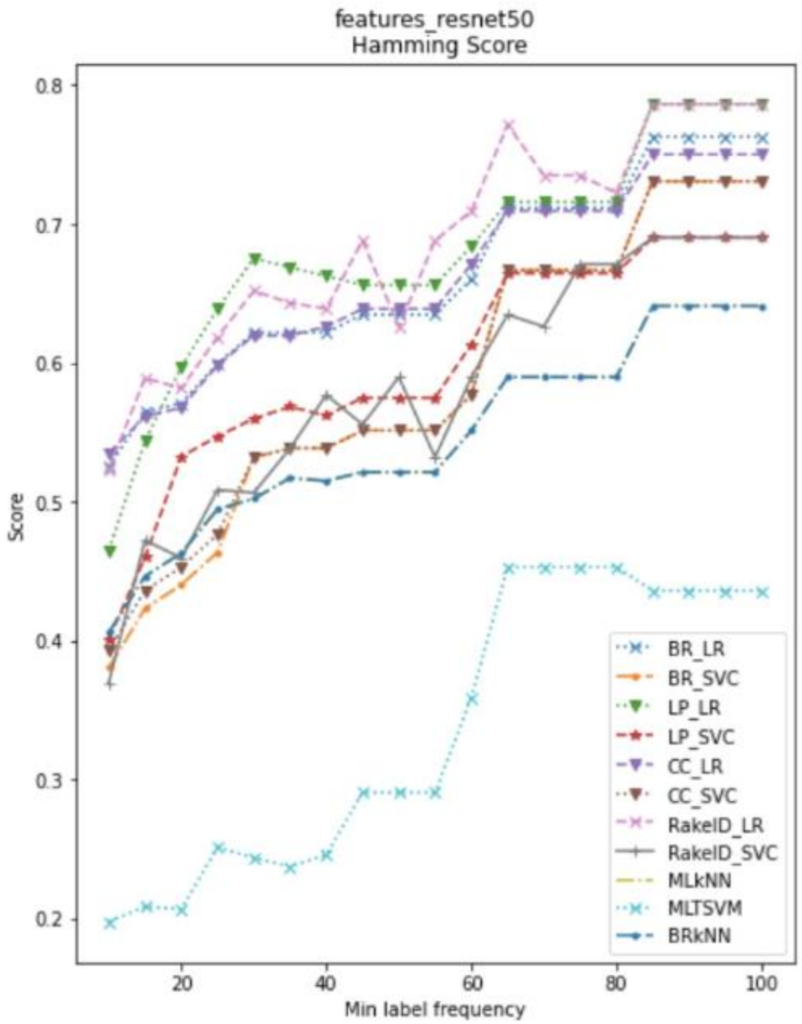


(b) Features from ResNet50

7. Results



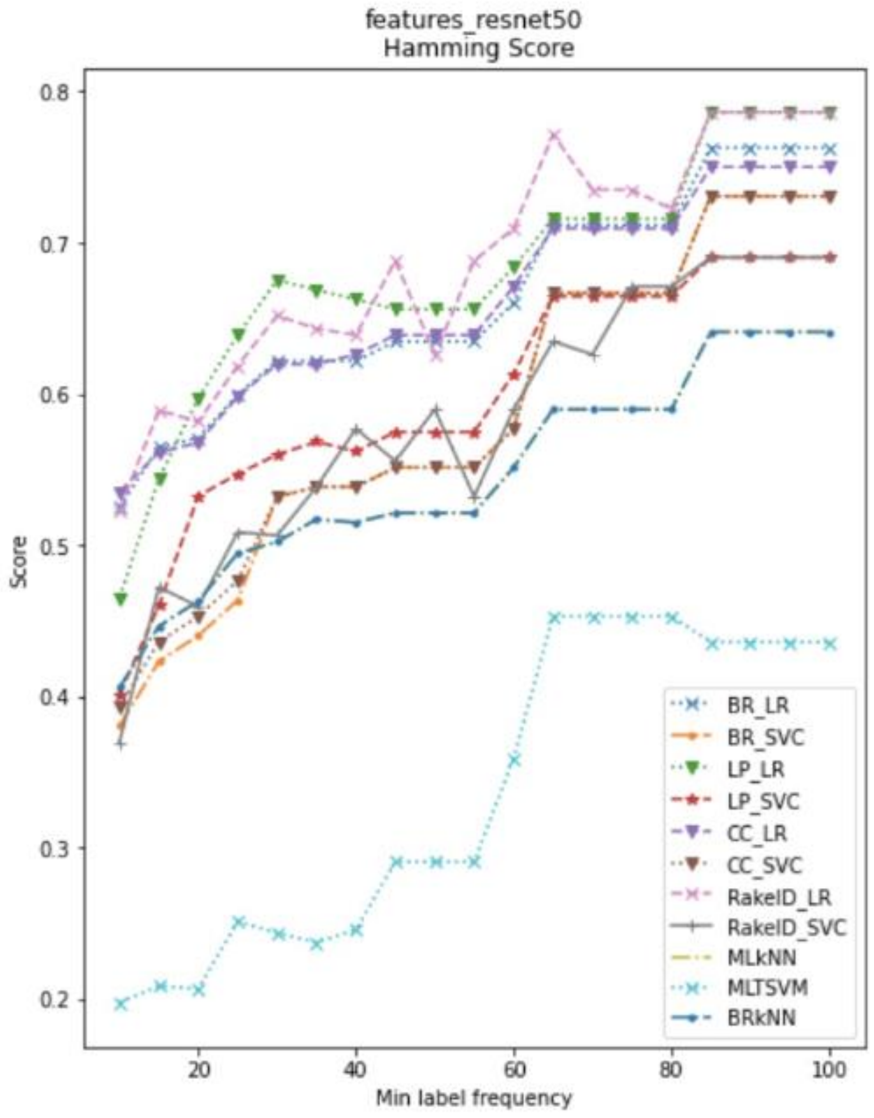
(a) Features from ResNet18



(b) Features from ResNet50

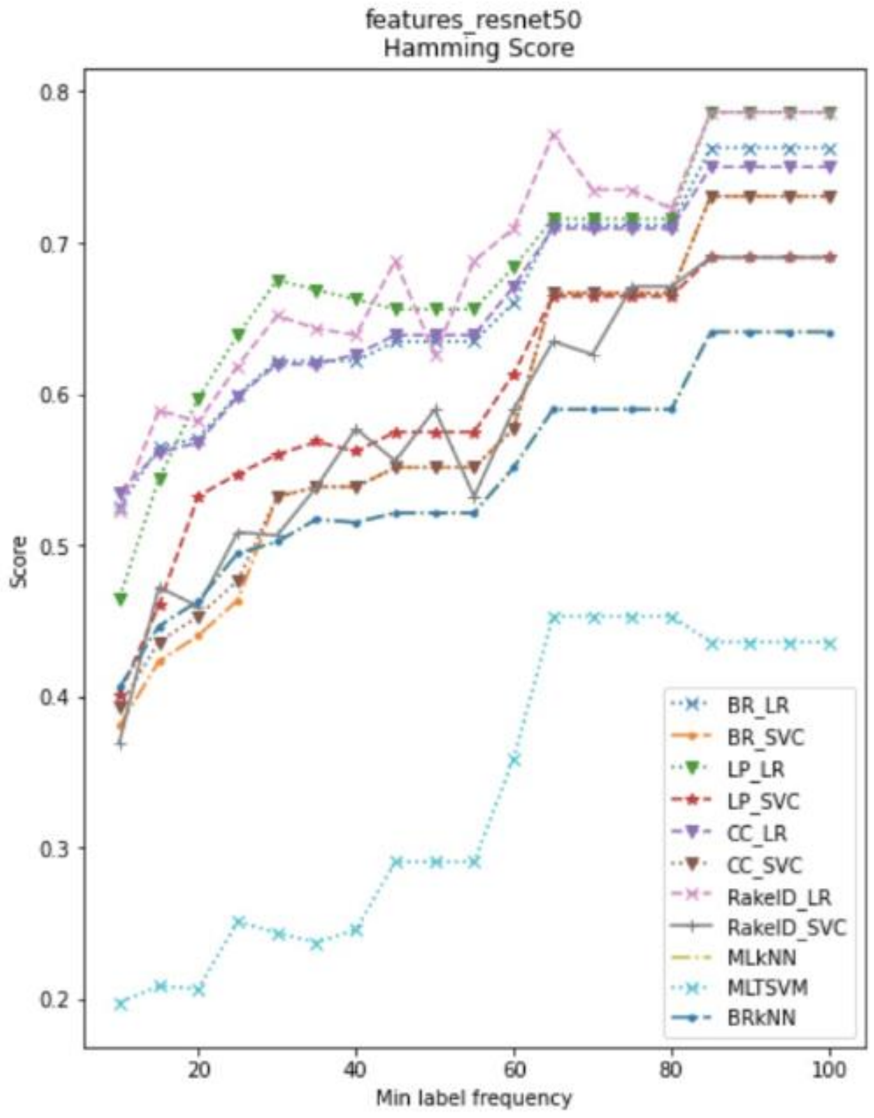
8. Discussion

- Firstly, we note that as we move along the abscissa, the methods tend to give better results.



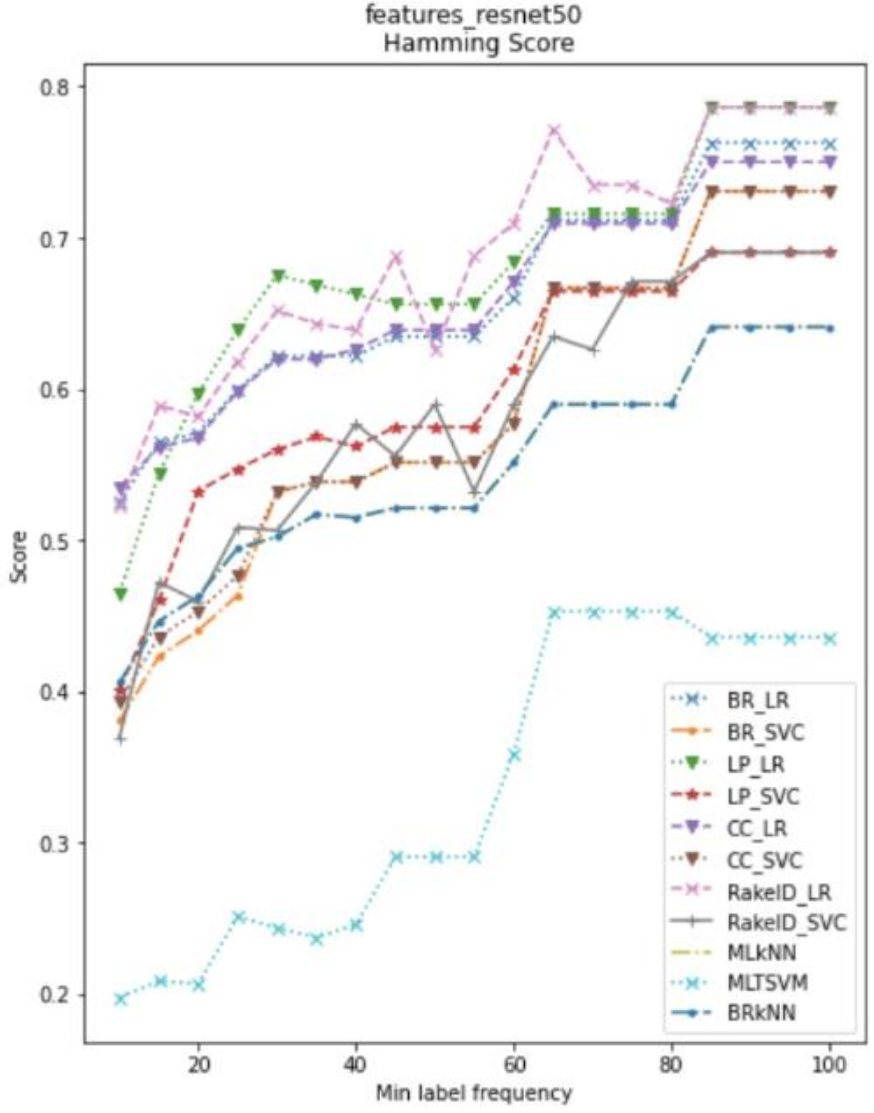
8. Discussion

- Firstly, we note that as we move along the abscissa, the methods tend to give better results.
- This is reasonable: the higher the threshold, less labels to predict, and more examples per label.



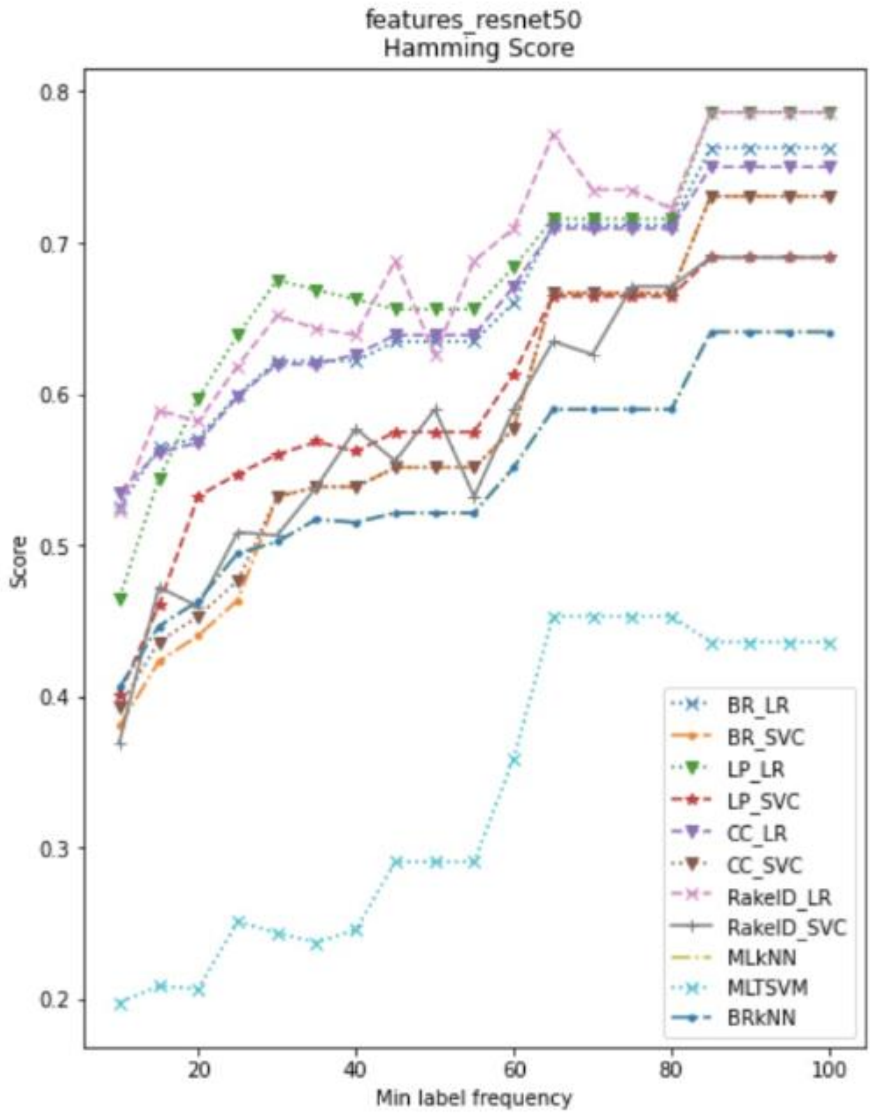
8. Discussion

- Firstly, we note that as we move along the abscissa, the methods tend to give better results.
- This is reasonable: the higher the threshold, less labels to predict, and more examples per label.
- We also note that the Problem Transformation Methods give better results than Algorithm Adaptation ones.



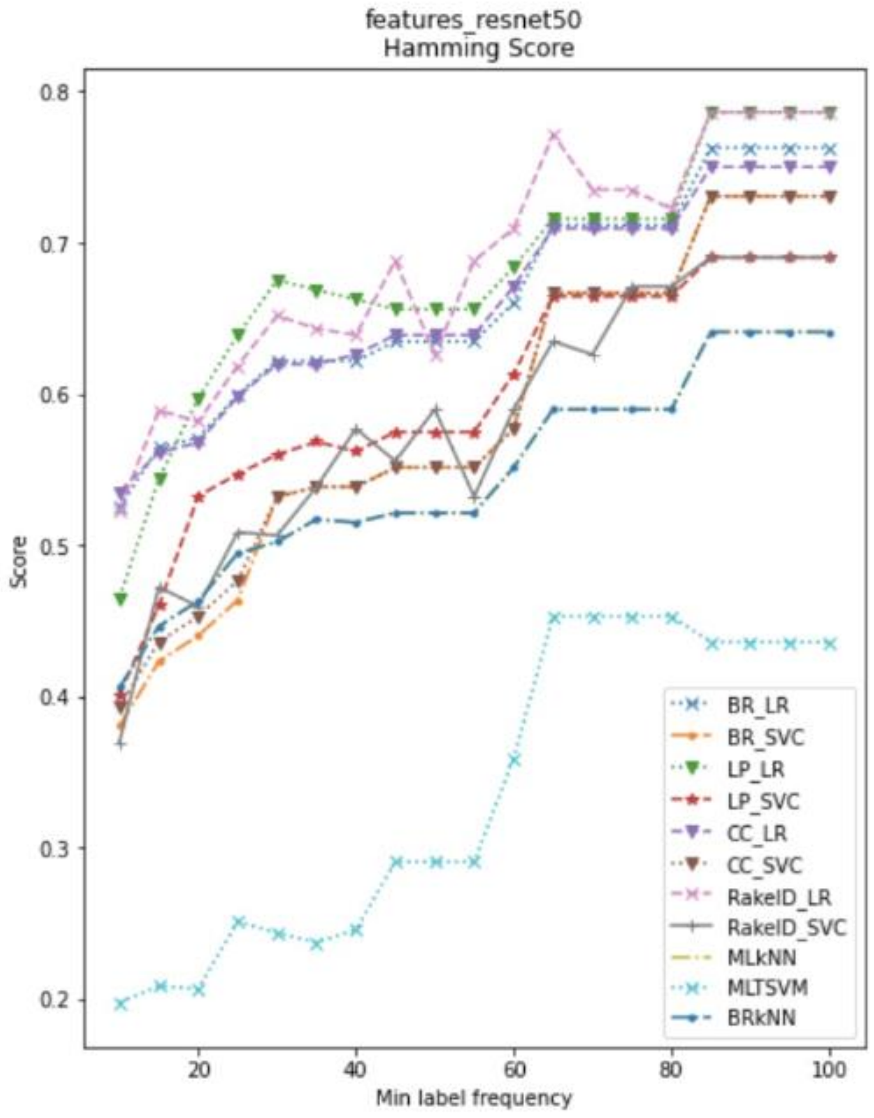
8. Discussion

- We also observe that ResNet50 descriptors gives rise to better results than the ones from ResNet18.



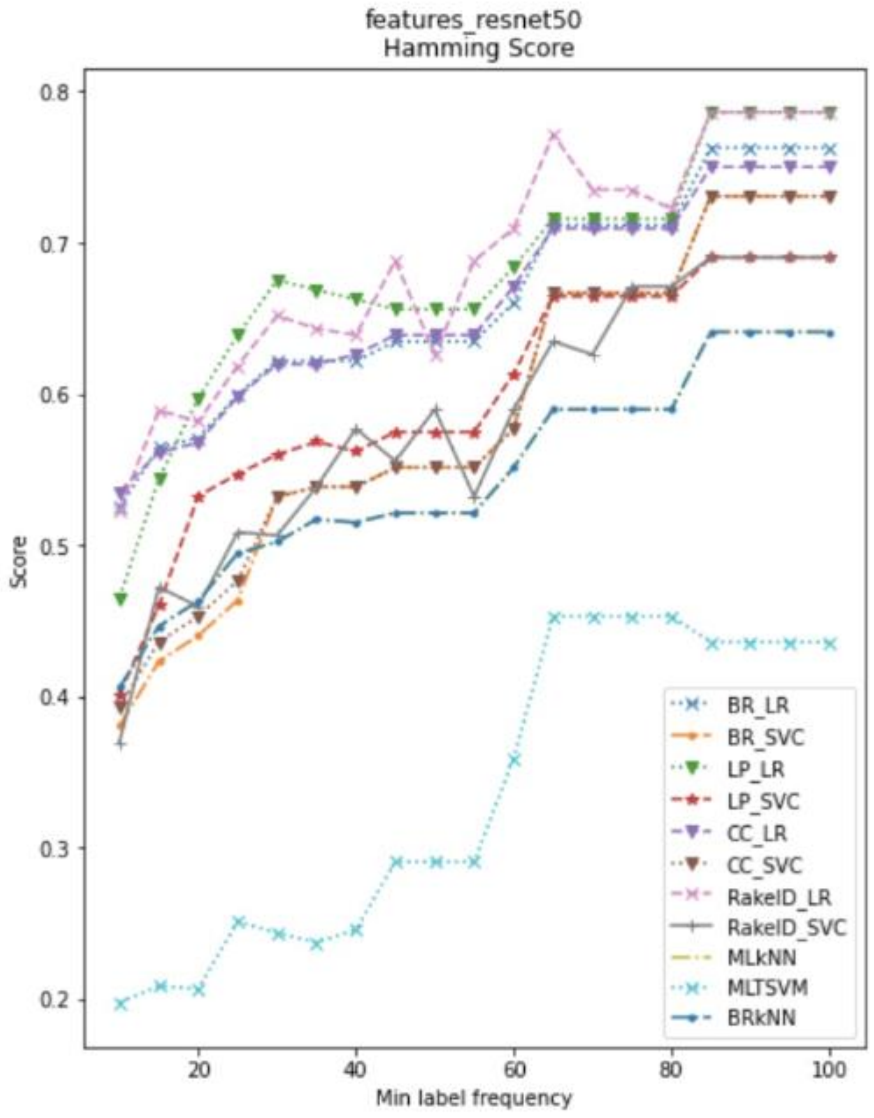
8. Discussion

- We also observe that ResNet50 descriptors gives rise to better results than the ones from ResNet18.
- This is especially noticeable for low threshold values.



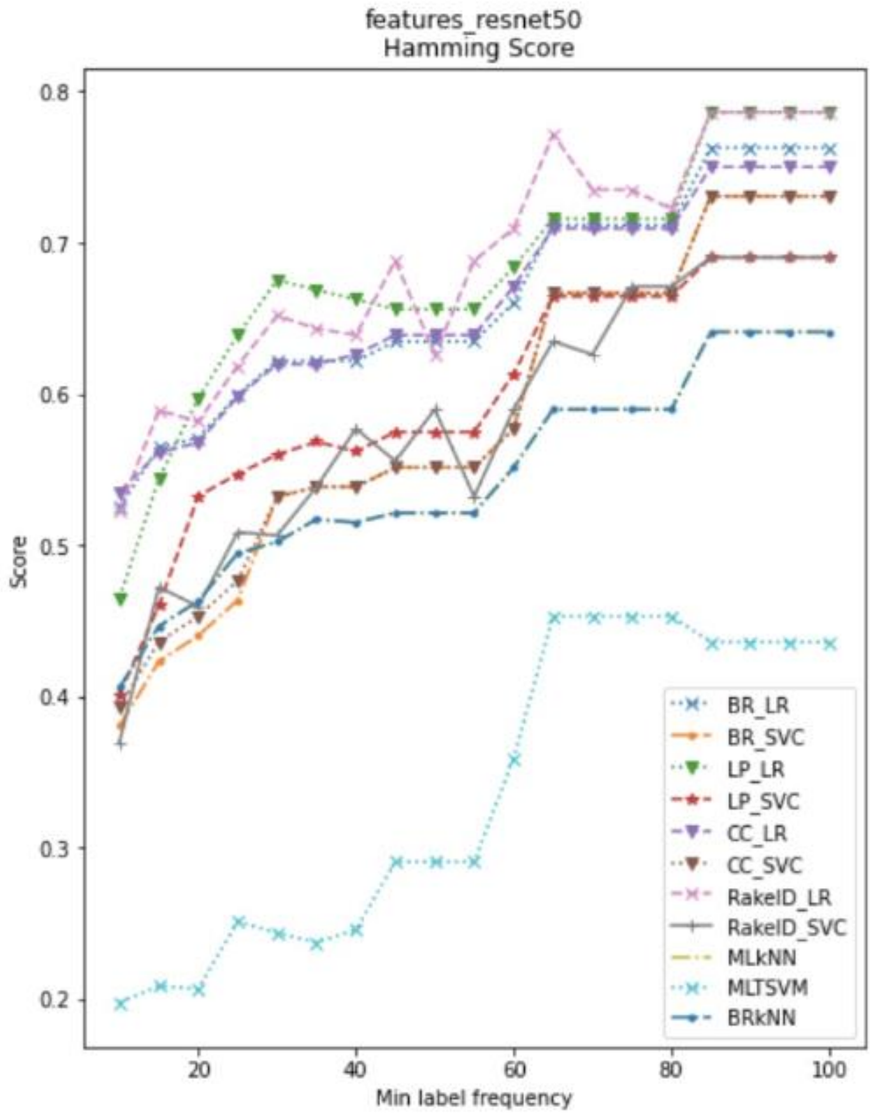
8. Discussion

- We also observe that ResNet50 descriptors gives rise to better results than the ones from ResNet18.
- This is especially noticeable for low threshold values.
- We want to keep the problem as interesting as possible, i.e., use the lowest threshold possible.



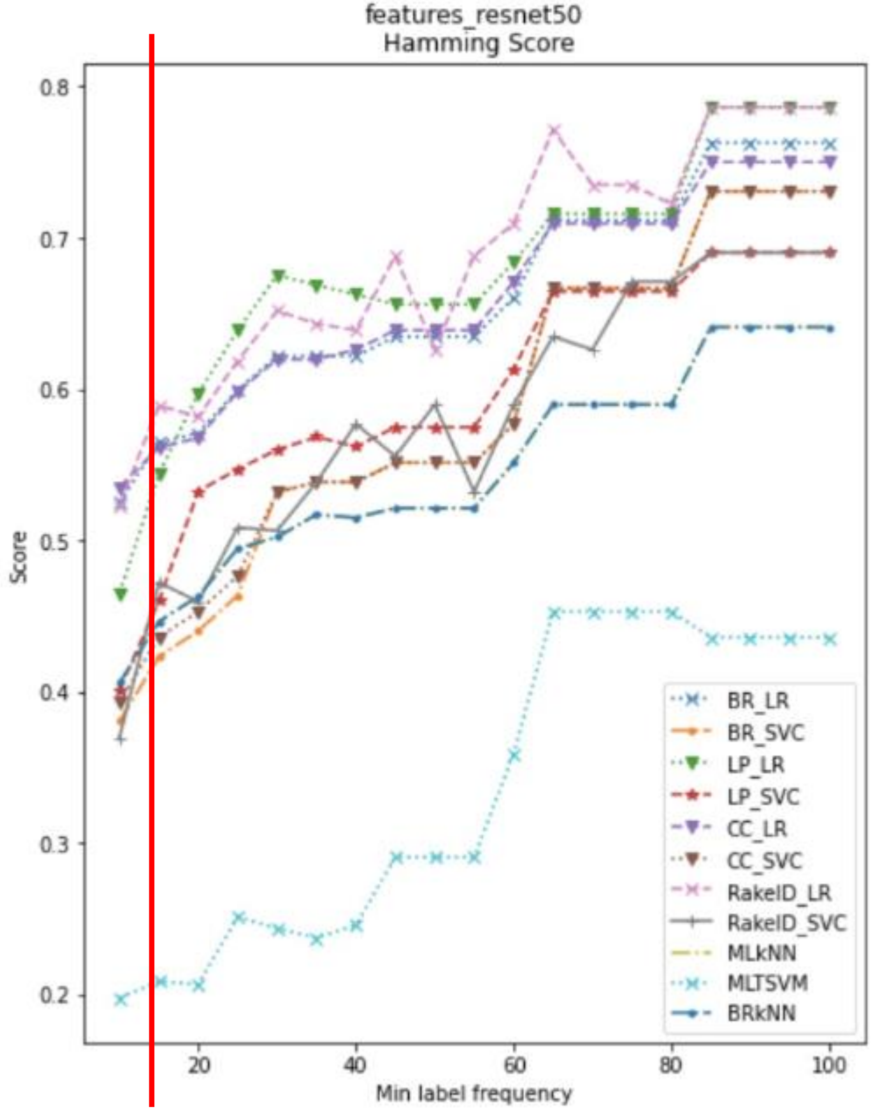
8. Discussion

- We also observe that ResNet50 descriptors gives rise to better results than the ones from ResNet18.
- This is especially noticeable for low threshold values.
- We want to keep the problem as interesting as possible, i.e., use the lowest threshold possible.
- Let's assume that $t = 15$ is enough. This left us with the 26 most frequent labels (4.43% of total labels)



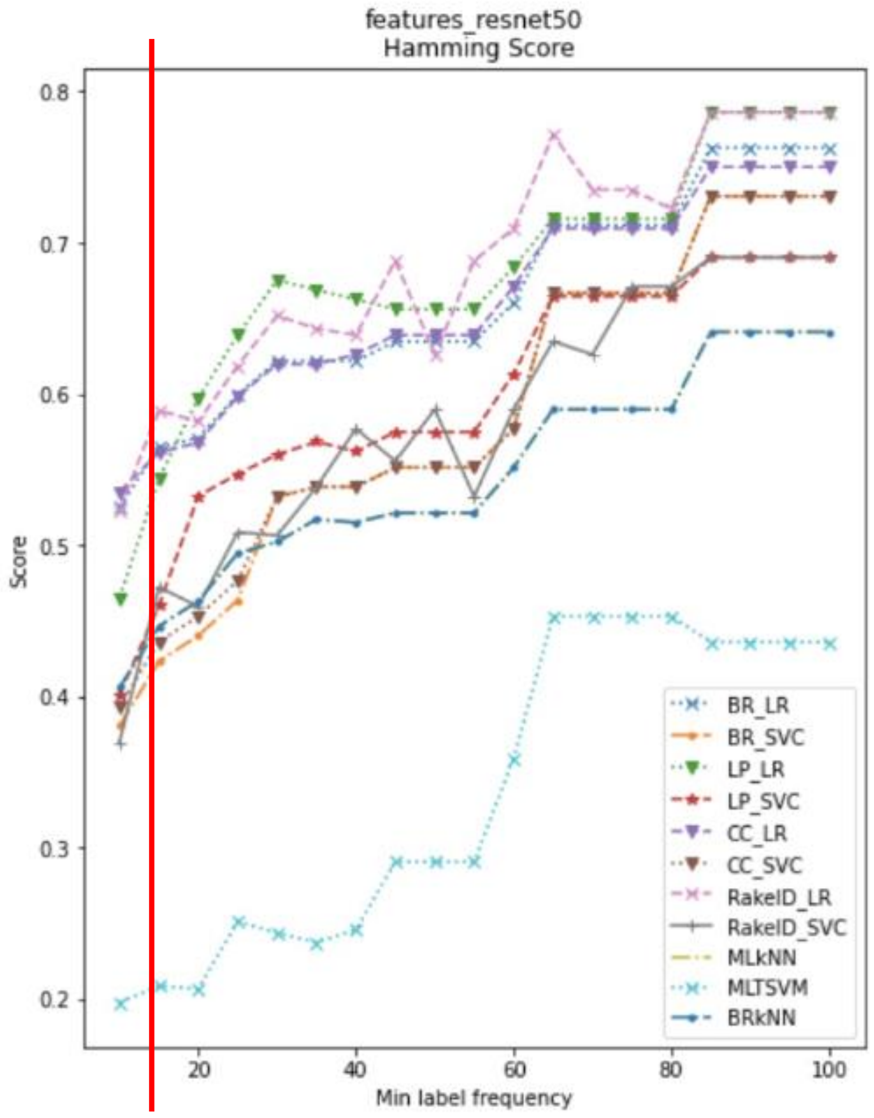
8. Discussion

- In this situation, we notice that some methods stand out above the rest.



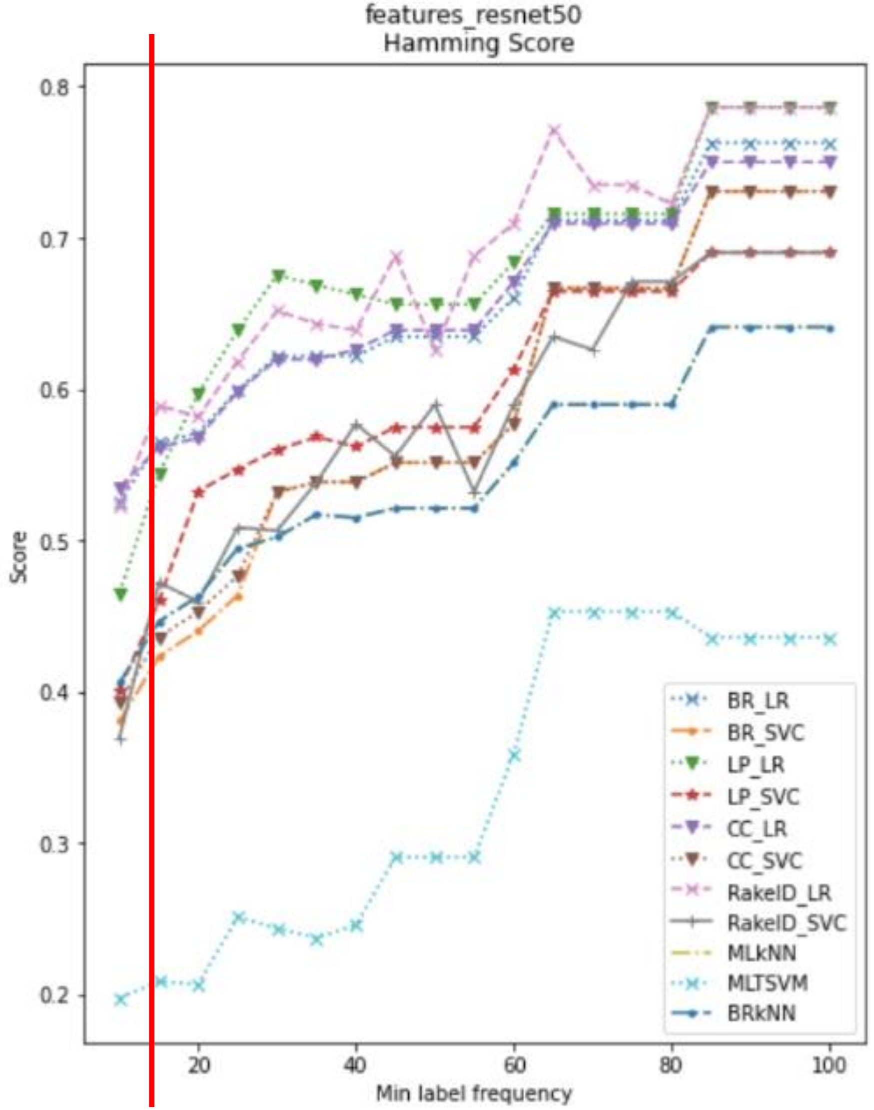
8. Discussion

- In this situation, we notice that some methods stand out above the rest.
- These are, in decreasing order, RakelD_LR, CC_LR, BR_LR and LP_LR.



8. Discussion

- In this situation, we notice that some methods stand out above the rest.
- These are, in decreasing order, RakelD_LR, CC_LR, BR_LR and LP_LR.

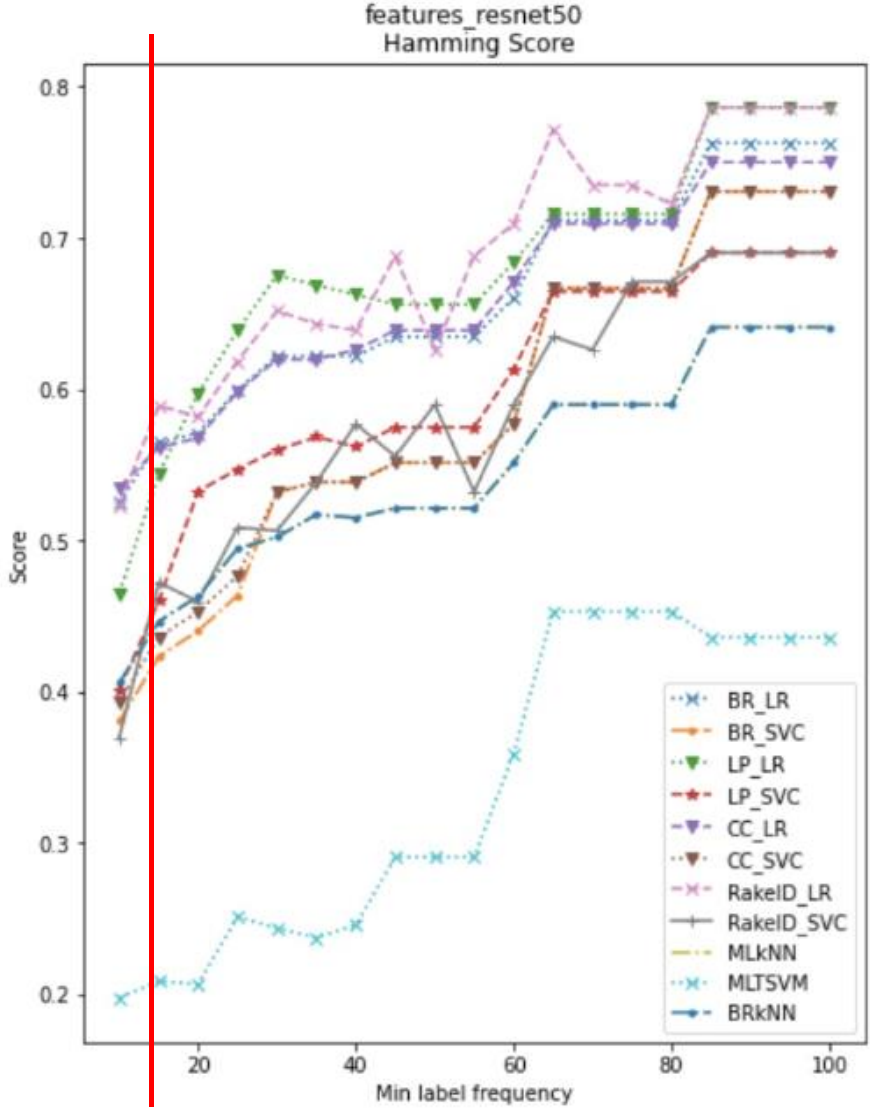


Method	Exact Match Ratio	Hamming Loss	Hamming Score
RakelD_LR	0.452	0.034	0.589
CC_LR	0.437	0.035	0.561
BR_LR	0.417	0.035	0.564
LP_LR	0.392	0.042	0.543

8. Discussion

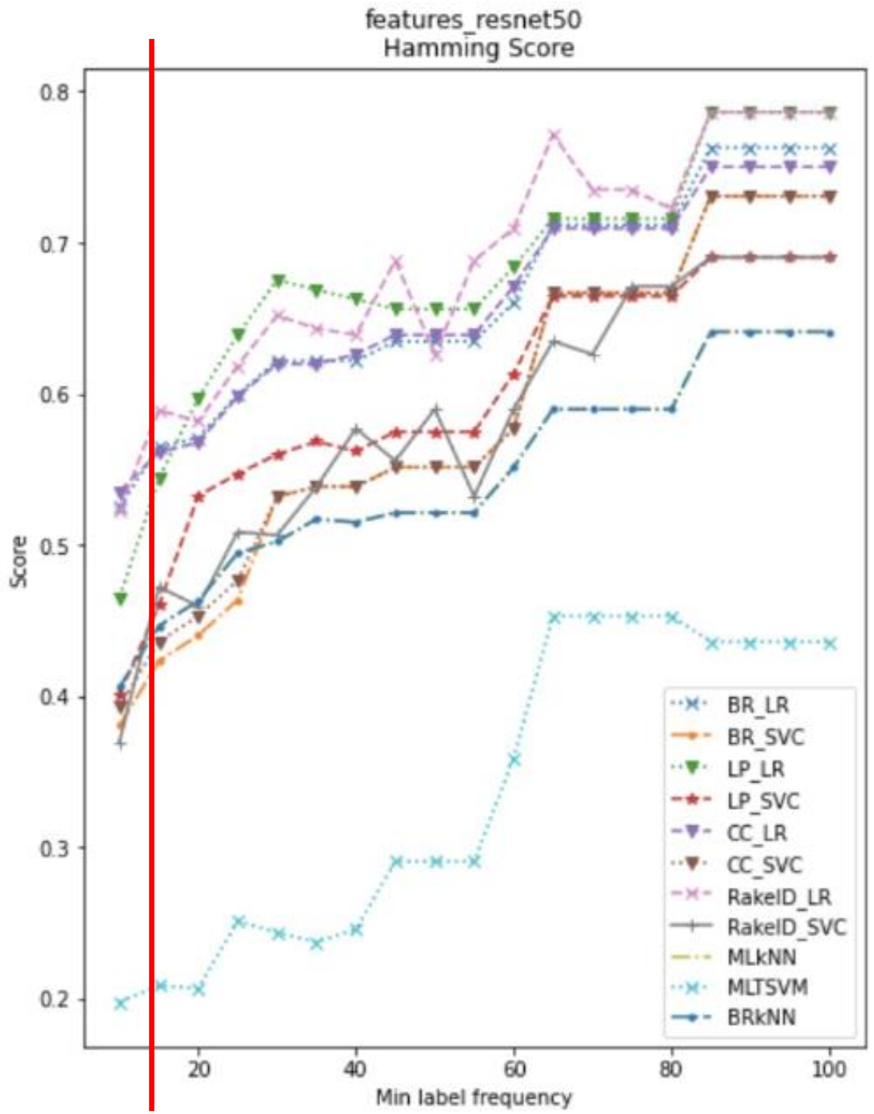
- In this situation, we notice that some methods stand out above the rest.
- These are, in decreasing order, RakelD_LR, CC_LR, BR_LR and LP_LR.
- From all of them, RakelD_LR obtains the best results for all metrics.

Method	Exact Match Ratio	Hamming Loss	Hamming Score
RakelD_LR	0.452	0.034	0.589
CC_LR	0.437	0.035	0.561
BR_LR	0.417	0.035	0.564
LP_LR	0.392	0.042	0.543



8. Discussion

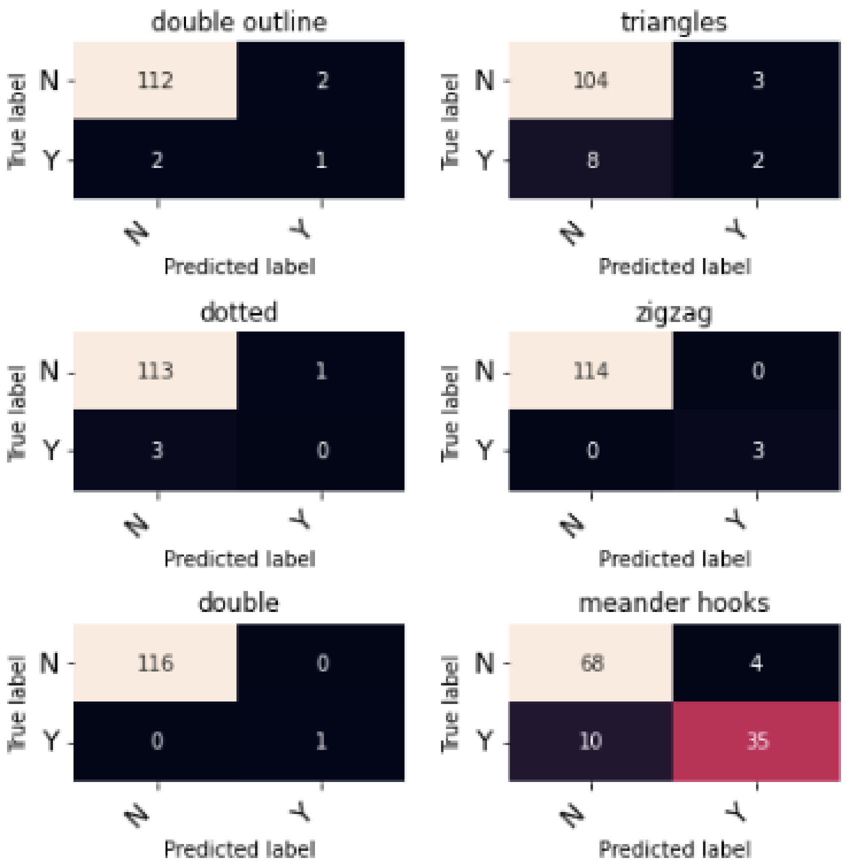
- In this situation, we notice that some methods stand out above the rest.
- These are, in decreasing order, RakelD_LR, CC_LR, BR_LR and LP_LR.
- From all of them, RakelD_LR obtains the best results for all metrics.
- It is then reasonable to study it's results in depth.



Method	Exact Match Ratio	Hamming Loss	Hamming Score
RakelD_LR	0.452	0.034	0.589
CC_LR	0.437	0.035	0.561
BR_LR	0.417	0.035	0.564
LP_LR	0.392	0.042	0.543

8. Discussion

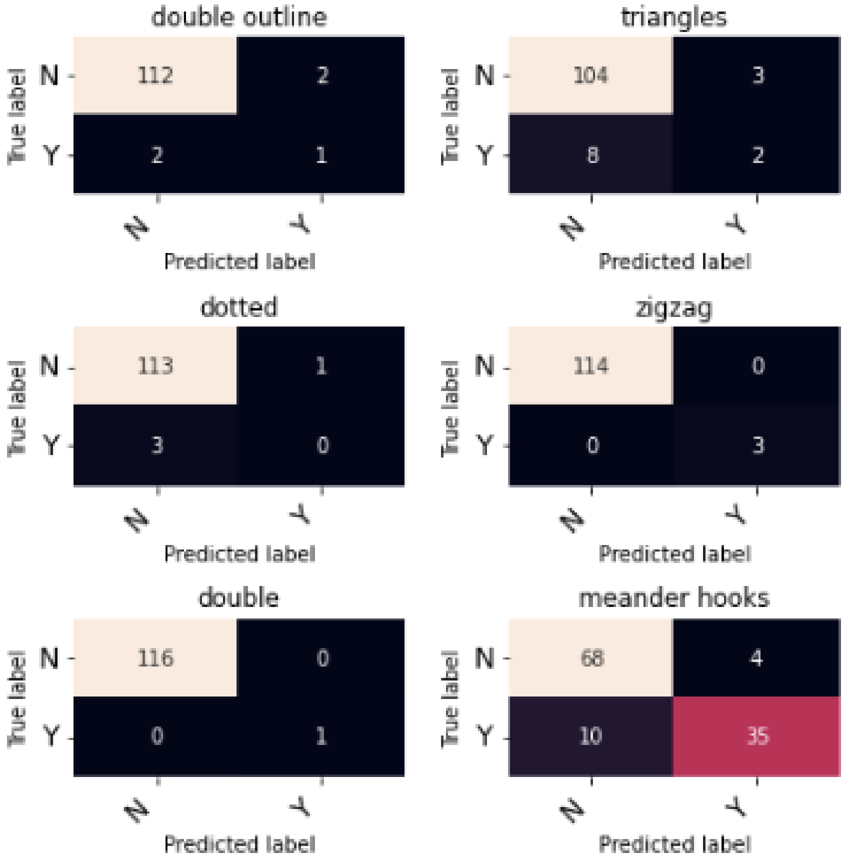
To get a better understanding of the results, we study them through confusion matrices.



8. Discussion

To get a better understanding of the results, we study them through confusion matrices.

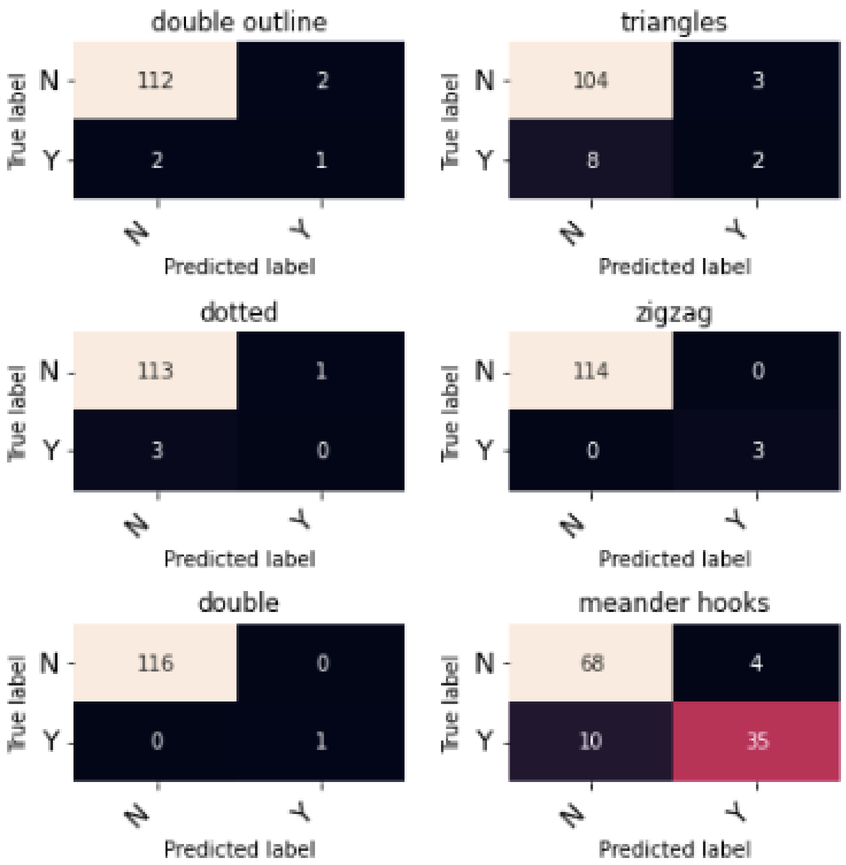
- We note that approximately 95% of classifications fall into true negatives.



8. Discussion

To get a better understanding of the results, we study them through confusion matrices.

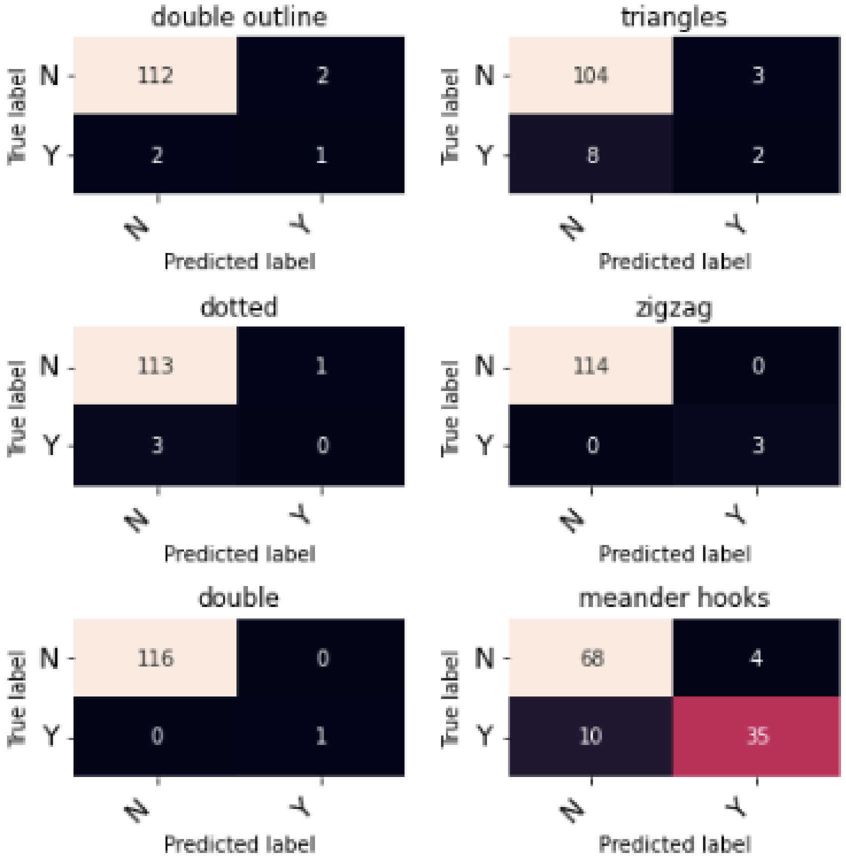
- We note that approximately 95% of classifications fall into true negatives.
- This shows an exaggerated disproportion between negative and positive cases for each label. We assume this happens both in test and training.



8. Discussion

To get a better understanding of the results, we study them through confusion matrices.

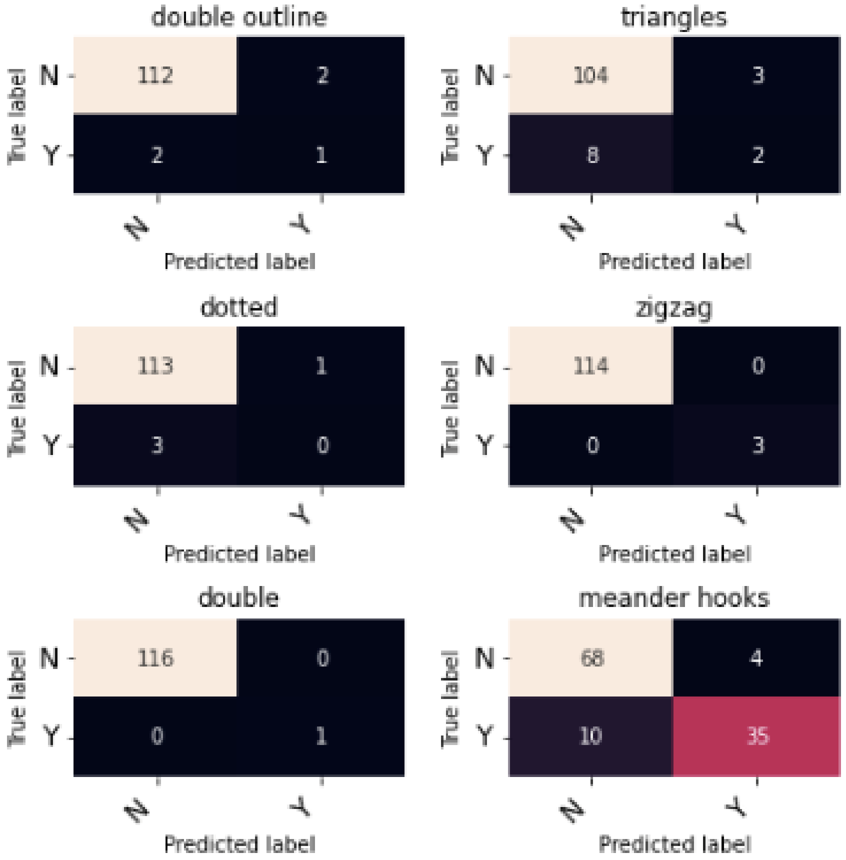
- We note that approximately 95% of classifications fall into true negatives.
- This shows an exaggerated disproportion between negative and positive cases for each label. We assume this happens both in test and training.
- As consequence, the algorithm is managing to **learn correctly when a pattern should not be tagged with a certain label,** but **not in the opposite case**: when the label correspond to that pattern.



8. Discussion

There are many possible reasons for this:

- The already mentioned low label density,
- A poor construction of training and test set.

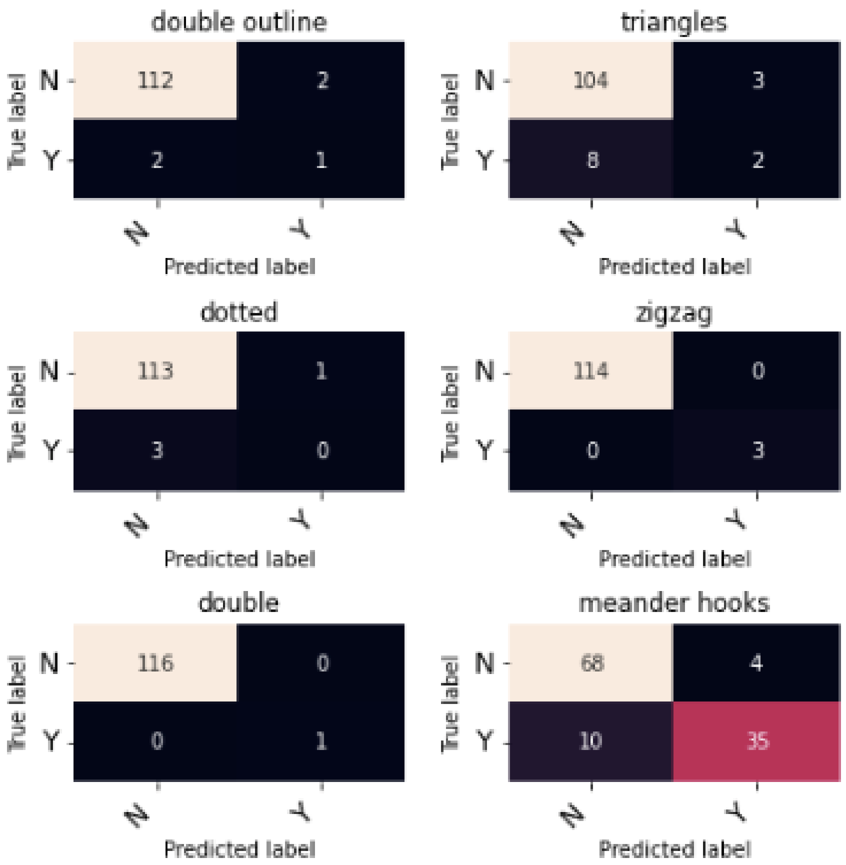


8. Discussion

There are many possible reasons for this:

- The already mentioned low label density,
- A poor construction of training and test set.

However, since we know that label density is extremely low, we can assume that the difficult lies there. This way, the problem is in the data set itself.



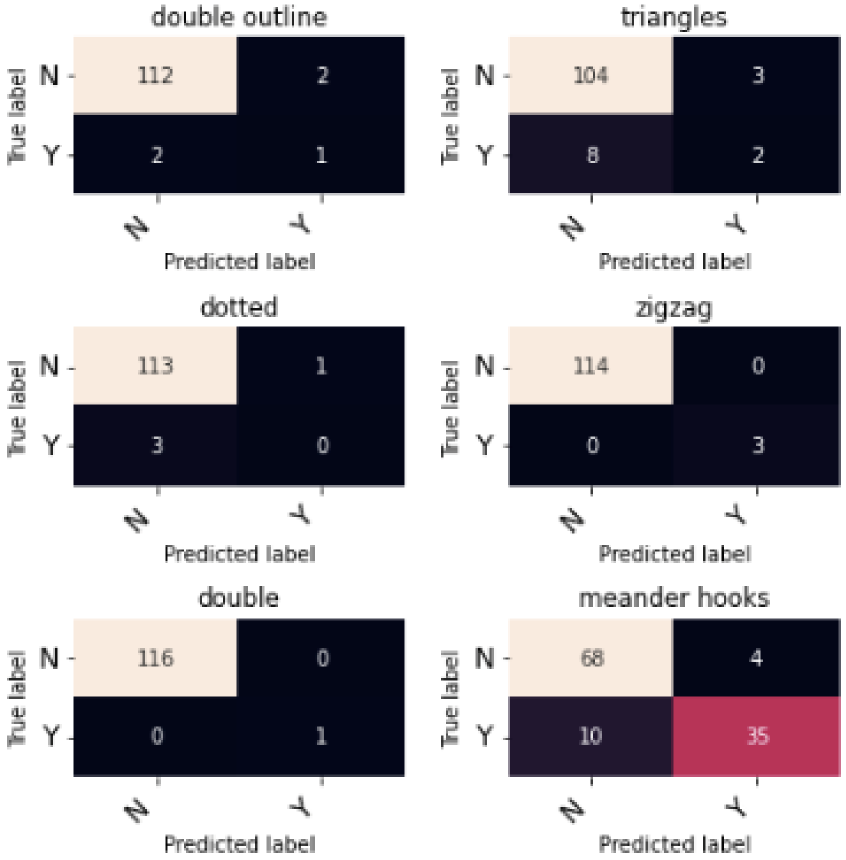
8. Discussion

There are many possible reasons for this:

- The already mentioned low label density,
- A poor construction of training and test set.

However, since we know that label density is extremely low, we can assume that the difficult lies there. This way, the problem is in the data set itself.

Too many labels, and only a few of them for each pattern. As consequence, too many negative examples, very few positives.



8. Discussion

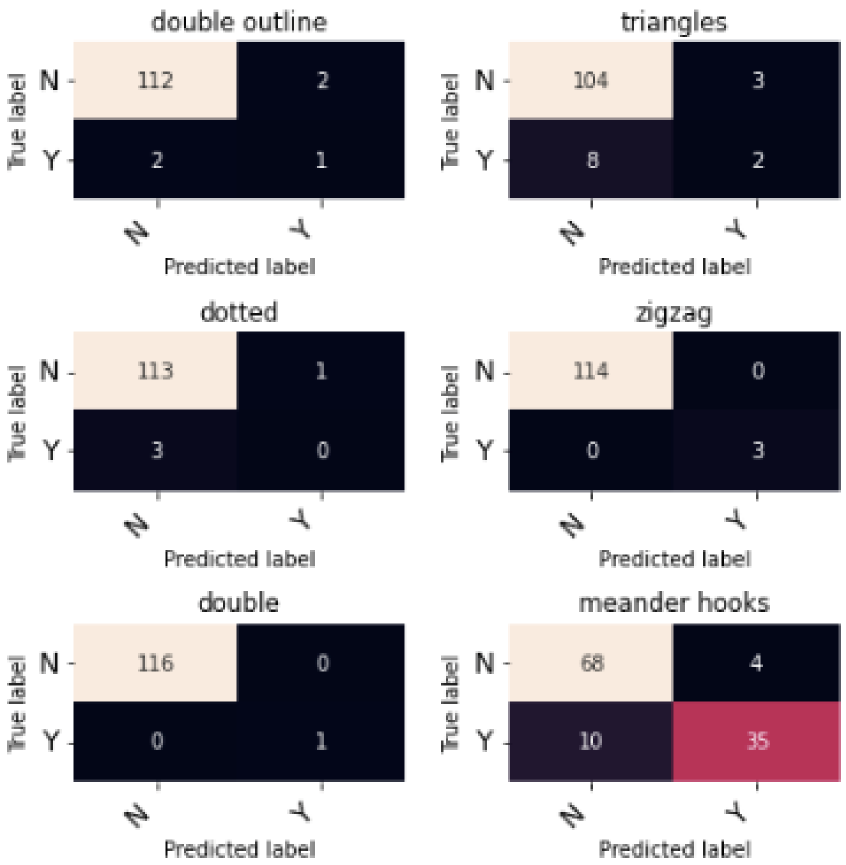
There are many possible reasons for this:

- The already mentioned low label density,
- A poor construction of training and test set.

However, since we know that label density is extremely low, we can assume that the difficult lies there. This way, the problem is in the data set itself.

Too many labels, and only a few of them for each pattern. As consequence, too many negative examples, very few positives.

...But there are some steps that we can take to deal with it.



9. Future work

We detected that one way to improve the data set is by **homogenizing the labels**. This to treat some undesirable cases:

- Labels with stop words, such as “**as** filling ornament”
- Patterns with plural labels, which do not include individual ones. For example, patterns with the label “triangle**s**” but not with “triangle”.

9. Future work

We detected that one way to improve the data set is by **homogenizing the labels**. This to treat some undesirable cases:

- Labels with stop words, such as “**as** filling ornament”
- Patterns with plural labels, which do not include individual ones. For example, patterns with the label “triangle**s**” but not with “triangle”.

In this regard, we consider that stemming or lemmatisation could be good alternatives to deal with this problem from an algorithmic approach.

- However, it is also recommended to include the opinion of experts, to understand what characteristics of the labeling we should respect and which we can modify.

9. Future work

We detected that one way to improve the data set is by **homogenizing the labels**. This to treat some undesirable cases:

- Labels with stop words, such as "**as** filling ornament"
- Patterns with plural labels, which do not include individual ones. For example, patterns with the label "triangle**s**" but not with "triangle".

In this regard, we consider that stemming or lemmatisation could be good alternatives to deal with this problem from an algorithmic approach.

- However, it is also recommended to include the opinion of experts, to understand what characteristics of the labeling we should respect and which we can modify.

This way we could reduce the number of single-event labels, as well as increasing the number of labels per pattern.

10. Conclusions

The work carried out gives rise to several conclusions. between them we have:

10. Conclusions

The work carried out gives rise to several conclusions. between them we have:

- **The superiority of ResNet50 descriptors over the ones from ResNet18.**
  - This also leaves a question: will we still have better results if we increase the dimensionality? Up to what point?

10. Conclusions

The work carried out gives rise to several conclusions. between them we have:

- **The superiority of ResNet50 descriptors over the ones from ResNet18.**
  - This also leaves a question: will we still have better results if we increase the dimensionality? Up to what point?
- **Problem Transformation Methods over Algorithm Adaptation ones.**
  - Again, with the conclusion comes the question: why does this phenomenon happen? Which feature of the data set is best covered by the problem transformation?

10. Conclusions

The work carried out gives rise to several conclusions. between them we have:

- **The superiority of ResNet50 descriptors over the ones from ResNet18.**
  - This also leaves a question: will we still have better results if we increase the dimensionality? Up to what point?
- **Problem Transformation Methods over Algorithm Adaptation ones.**
  - Again, with the conclusion comes the question: why does this phenomenon happen? Which feature of the data set is best covered by the problem transformation?
- **The need to define to what level we will prune the problem.**
  - Bearing in mind that the more we can, the easier the problem is… But less interesting as well.

10. Conclusions

The work carried out gives rise to several conclusions. between them we have:

- **The superiority of ResNet50 descriptors over the ones from ResNet18.**
  - This also leaves a question: will we still have better results if we increase the dimensionality? Up to what point?
- **Problem Transformation Methods over Algorithm Adaptation ones.**
  - Again, with the conclusion comes the question: why does this phenomenon happen? Which feature of the data set is best covered by the problem transformation?
- **The need to define to what level we will prune the problem.**
  - Bearing in mind that the more we can, the easier the problem is… But less interesting as well.
- **Investigate techniques and apply them to labels in order to improve the data set.**
  - We need to increase the label density to get better results.

Kunisch Recognition through Multi-Label Classification Algorithms

Prof. Benjamín Bustos

Prof. Iván Sipirán

Matías Vergara