PetFinder.my - Pawpularity

Armando Fortes[1], **David Pissarra**[2]
Machine Learning Course Fall 2021
Department of Computer Science and Technology
Tsinghua University, Beijing, China
{ferreiracardos10[1], pissarrad10[2]}@mails.tsinghua.edu.cn

Abstract

Pawpularity is a metric created by *PetFinder.my*, Malaysia's leading animal welfare platform, which aims to measure pet photos' popularity. Nowadays, millions of stray animals are alone on the streets or living at shelters, waiting for a new home. Therefore, by taking advantage of the data science and machine learning fields, we may be able to predict the impact of certain features in pet photos, which might help them get more attention from the population. Consequently, we would help improve their chances of being rescued and finding loving homes.

1 Introduction

The significant number of stray animals around the world justifies the recent effort made by humans to find better conditions for these pets. Therefore, there has emerged a necessity for building new mechanisms and methods to generate more interest from the population in giving them new homes. With the development and application of the right machine learning techniques, we may be able to design a model which, given a picture of any kind of pet, will predict how popular it would be. Accordingly, the popularity of a homeless pet's photo is directly related to the visibility that it should receive. Moreover, a popular photo will have more attention from the public and, consequently, increase the chances of this pet being adopted. Also, the model may help identify the features and patterns which will lead to a popular photo, creating the opportunity for future pictures to have a better outcome than the previous ones. Having a first look at the Pawpularity Contest [1] and respective dataset, we can formalize it as a supervised learning problem. Regarding this contest, a large dataset is given for both training and testing purposes, which contains various picture of domestic animals. Also, attached to each image there is a set of metadata with various features and most importantly the correspondent *Pawpularity*, of course the latter is only available in the training set. Although it may also be modeled by many other machine learning approaches, as we are analysing images, we are encouraged to use some computer vision techniques such as Convolutional Neural Networks (CNN) or Transformer models based on self-attention. These types of models have represented a huge breakthrough in image analysis, being the reason for our initial intuition.

2 Related Work

2.1 *PetFinder.my*'s Cuteness Meter

PetFinder.my has been developing an algorithm - *Cutness Meter* [2] - related to this contest, which rates the cutness of a pet photo with 1 to 5 stars. The current application for this algorithm helps *PetFinder.my*'s users receive feeback from the photos they upload. Thus, they may be able to understand actually how attractive their photos are and which one they definitely can improve. This algorithm is still in an experimental stage, and this is one of the motivations for the establishment of this contest. Since this is a competition, the teams who create the best performing algorithms may be invited to contribute to this cause by implementing their solutions on Malaysia's leading animal welfare platform. Therefore, this is how we can relate *Pawpularity* with the *Cutness Meter*.

2.2 The evolution of Machine Learning for Computer Vision

Convolutional Neural Networks were first introduced in the 1980s, by Yann LeCun, in an attempt to distinguish and classify handwritten digits - *LeNet*. In spite of presenting a great breakthrough in image recognition, there still were some issues regarding the scaling of this method, due to its computational complexity and need for large amounts of data. Only by the year of 2012, were we able to achieve the first strong results with CNNs. Alex Krizhevsky et al. [3] presented an improved CNN architecture - *AlexNet* - to classify the 1.2 million high-resolution images from the ImageNet challenge into the 1000 different labels. The main techniques introduced by the *AlexNet* which made this possible were the use of *Dropout* to prevent overfitting, the use of the *ReLU* activation function, which showed improved training performance over *tanh* and *sigmoid*, and also having a deeper and wider architecture. Therefore, it represents an important milestone on the evolution of the computer vision field, serving as a base for the following developed CNNs. Some notable examples of successful implementations are the *GoogLeNet* [4], the *ResNet* [5], the *DenseNet* [6] and many others [7, 8, 9, 10]. Moreover, state of the art results continuously keep being achieved with various new creative approaches and architectures. Accordingly, due to the success of self-attention models, such as Transformers, in natural language processing, novel works have presented self-attention techniques to overcome the limitations presented by inductive convolutional biases [11, 12]. One of the newly developed Transformers, able to achieve SOTA results, was the Swin-Transformer [13].

2.3 Latest State of the Art results in Image Recognition

Since the breakthrough of *AlexNet*, Convolutional Neural Networks have been the dominating model architecture for computer vision. Many new architectures have been performing even better, compared to previous techniques. Dai et al. [14] combined both CNNs and self-attention within their model together to form a complete network, which is presented as *CoAtNet*. The latest implementation of this network - *CoAtNet-7* - was able to achieve 90.88% top-1 accuracy on ImageNet, with JFT-3B Google dataset, which consists of nearly 3 billion images, annotated with a class-hierarchy of around 30k labels. Taking advantage of existing neural networks, Xie et al. [15] were able to train an *EfficientNet* model [16] on labeled ImageNet images and use it as a 'teacher' to generate pseudo labels on JFT-300M Google dataset. Then, a 'student' model could be trained on a larger *EfficientNet*, combining labeled and pseudo labeled images, which were inferred previously by the 'teacher' model on unlabeled data. This process is iterated many times and each student model should later become a teacher model. Hence, student models will be subject to noise, so that the student generalizes better than the teacher. This self-training method was able to reach 88.4% currently top-13 accuracy on ImageNet. Also, Xie et al. [17] created another method – *Meta Pseudo Labels* - based on the previous one. However, unlike the last self-training method where the teacher is fixed, the teacher in *Meta Pseudo Labels* is constantly adapted by the feedback of the student's performance on the labeled dataset. Thus, the teacher will generate better pseudo labels to teach the student. This semi-supervised learning method achieved a better accuracy compared to the last method, conquering an accuracy of 90.2% on ImageNet.

2.4 Image Memorability and Interestingness

In recent years, the research community has started paying significant attention to popularity prediction in social media, which may be directed to all sorts of content such as messages, written posts, videos and images. Regarding the prediction of image popularity [18], it has been settled that one may leverage social cues (which tends to relate to the context behind the user and the publication or the influence that the publisher may have in the social media platform), as well as the intrinsic cues of the image (which commonly relates to the dominant colors of the image, contrast, texture, GIST [19] and so on). However, the content of an image can be considerably harder to extract and correlate with popularity, when compared to text-based content. Therefore, there has been an increasing interest in analyzing various semantic attributes of images, such as image interestingness [20] or image memorability [21, 22]. Context has once again been regarded as an aspect which largely contributes to these matters [20]. Moreover, with strong context, unusualness is the most important cue for interestingness. However, although it becomes more difficult to capture with weak contexts, there has also been evidence that there is a large degree of consistency among different viewers, and that some images are more interesting and memorable than others, even when there is no context or familiar elements (for example, famous monuments) [21]. Here, general preferences seem to take the main role. Furthermore, image quality and aesthetics [23] or relative spatial features [24] are some examples of attributes that have been recently explored in substantial detail. In terms of conclusion, by analyzing the presented semantic attributes of images, a better understanding has been developed for situations where one wants to extract the images which are more likely to be remembered by viewers from a given collection. This knowledge could be applied to selecting images for illustrations, covers, user interfaces, as well as for the promotion of stray animals for adoption, which is the main cause of this project.

3 Method

The main goal for our model is to predict the right *Pawpularity* of a given photo. This metric called *Pawpularity* is between 0 and 100, and is related to how popular a pet photo can be. Fundamentally, our model will be trained from a given dataset which is composed from more than 9000 pet photos, striving for accurate predictions.

3.1 Data Description

A dataset with more than 9000 pet photos is provided on this *Kaggle* competition. Each photo is labeled with its *Pawpularity*. In addition, photo metadata is also included in the dataset, which contains multiple extra binary features {0, 1} for each picture, such as:

- Focus - Pet is neither too close nor far.
- Eyes - At least 1 eye is visible (pupil should be clear).
- Face - Decently clear face.
- Near - Pet appears on significant portion of photo.
- Action - A pet action is detected, for instance jumping.
- Accessory - Photo contains either digital or physical accessory, for example digital stickers or physical toys.
- Group - More than 1 pet in the photo.
- Collage - Digitally-retouched photo.
- Human - Presence of humans in the photo.
- Occlusion - Undesirable objects blocking part of the pet.
- Info - Information in the photo by adding text or labels.
- Blur - Blurry part of the photo, especially for the pet's eyes and face. For blurry pictures, *Eyes* feature is always set to 0.

3.2 Metadata Insights

Our work was initiated in regards of analysing the given set of metadata features in order to have a better understanding on how each of the features may or may not affect *Pawpularity*. Accordingly, we plotted the *Pawpularity* frequency distribution and calculated the accordant mean for the total set of samples in the training set, as seen in Figure 1, and also for each value of the given metadata features, as seen in Figure 2.

From Figure 1, we are able to conclude that the vast majority of samples are located in the [20, 50] interval for *Pawpularity*, resulting in a mean value of 38.04. This could make it more difficult to generalize our inference, as there is such a high density of samples for a specific interval of *Pawpularity*. Regarding Figure 2, it is observable that for each metadata feature there is a significant discrepancy between the number of given samples for each value. For example, there are 9638 samples where *Subject Focus* has a value of 0 (which corresponds to 97.2% of the total number of samples) whereas only 274 samples have value 1 (resp. 2.8%). Similar unbalanced proportions can be found across the rest of the features, which is a sign that one may find it difficult to construct generalized knowledge from the given tabular data.

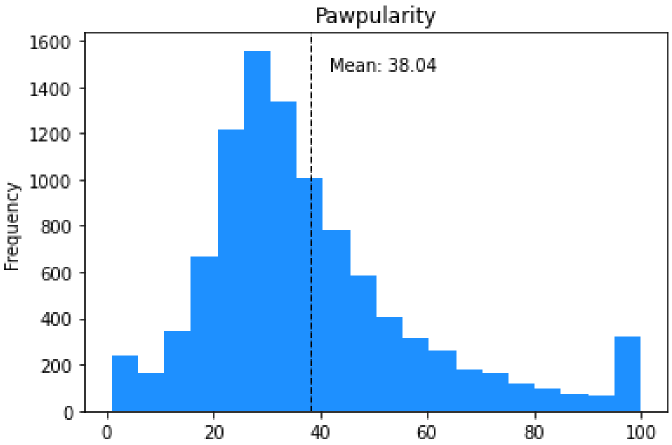


Figure 1: The *Pawpularity* distribution for the given training set.

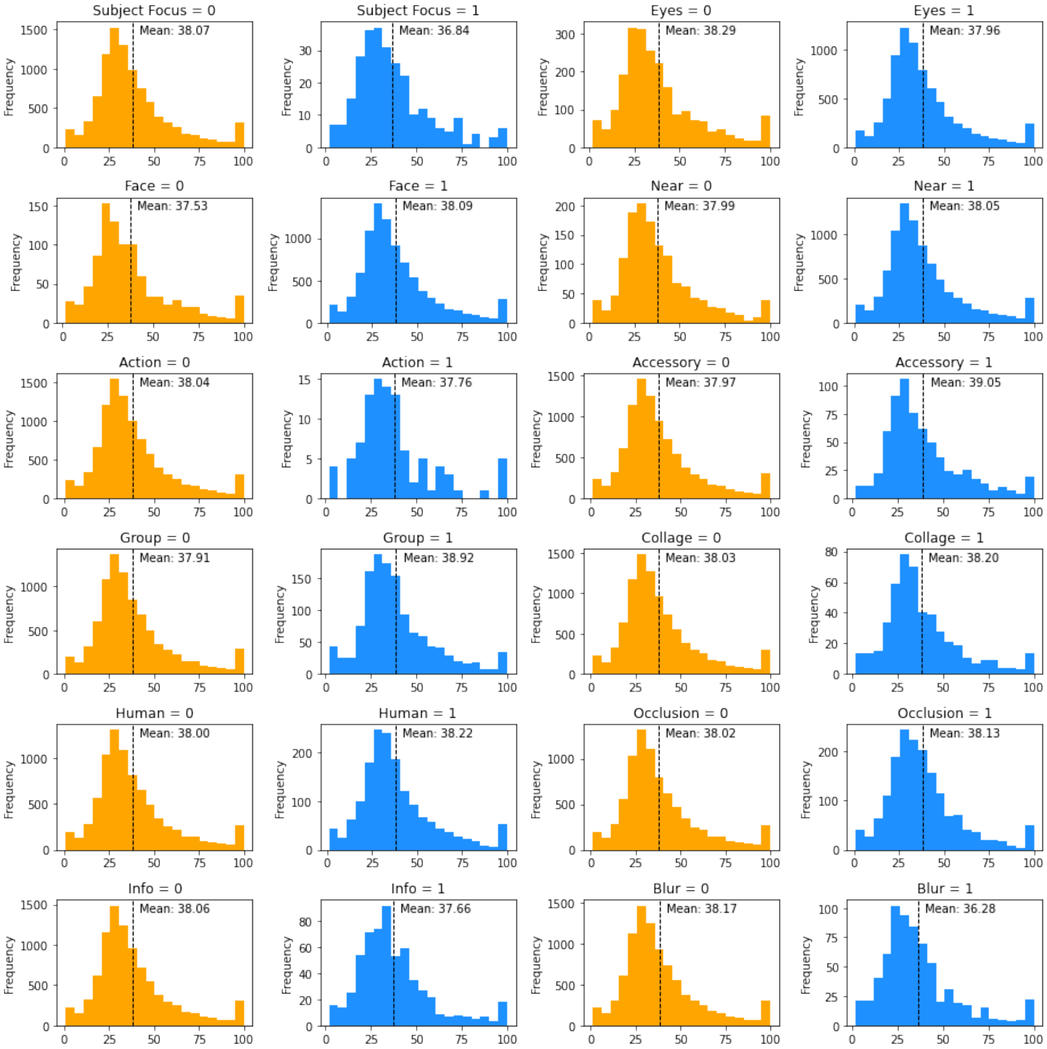


Figure 2: The *Pawpularity* distribution for each metadata feature value.

Moreover, it may also be noted that all the plotted distributions in Figure 2 considerably resemble the curve seen in the plotted distribution from Figure 1, as well as their identical mean values. These characteristics reiterate the hypothesis that the given metadata features may not be enough to discern whether the picture will be popular or not.
Furthermore, by gathering every metadata feature together, we were able to calculate the correlation between each of them and build the presented heatmap (Figure 3).

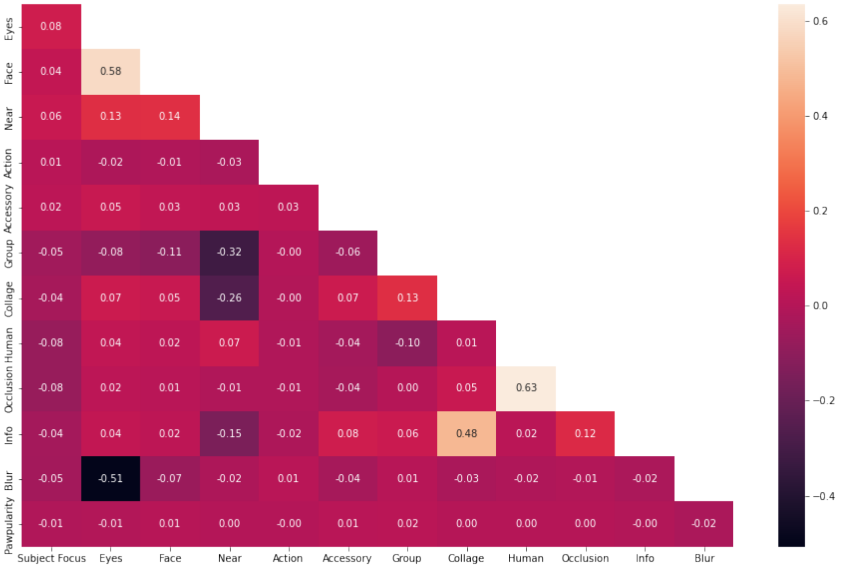


Figure 3: Correlation matrix regarding the metadata attributes.

In addition to the previous conclusions made, by analyzing the above correlation matrix between all the metadata attributes and the Pawpularity score, we understand again that there is very little connection between each feature and the desired label, which means the respective inference would have very high complexity.
Nevertheless, in order to further test our hypothesis, as well as potentially find combined connections between the different features, which may help extract some signals for the *Pawpularity* score, we decided to use the Gradient Boosting implementation XGBoost [25] (Section 3.4.2). This decision was made after several experiments with different models in an attempt to learn from the metadata features, being XGBoost the one with the best performance.

3.3 Data Pre-Processing

In order to build a model with improved performance, data pre-processing has an important role, since it allows the extraction of the most crucial features, as well as the removal of features and traits that can be harmful for the model, from the raw data given at the beginning.
Firstly, each picture is decoded into 3-Dimensional arrays of shape WxHxC, where W, H and C correspond to image width, image height and number of color channels, respectively. Since we will be working with Red, Green, Blue (RGB) pixel levels, there will be three channels ($C = 3$). Each entry will starts by being comprehended in [0,255] as it is the conventional interval for RGB pixel values. Subsequently, we proceed to perform normalization of our data where the resulting values are chosen to be in the range [0,1]. Data normalization is an important step where each parameter is ensured to have a similar data distribution, which has been shown to make convergence faster while training a network. Furthermore, in order to guarantee that the images have the same size, as it is required for most neural network models, each image is resized into a a square shape of dimensions 224x224, which means the final shape of the 3-Dimensional array of each picture is 224x224x3. The input will the correspond to a set of 9912 pictures the grouped into batches of size 32, where each picture is represented by the mentioned 3-D arrays. In regards of the labels, the given *Pawpularity* of each picture is also normalized into a range between [0, 1], since the given dataset provide scores between [0, 100].
Additionally, in order to increase the size of the dataset, we implemented data augmentation. To do

so, we adjusted some values, such as brightness, contrast, hue, and saturation. These augmentations were embedded inside the image batch, so the model can fetch augmentations during the learning process.

3.4 Model Description

The designed model for this Kaggle competition is essentially an Ensemble Model (Figure 4) from a neural network with a Swin transformer as the backbone, and a XGBoost regressor.

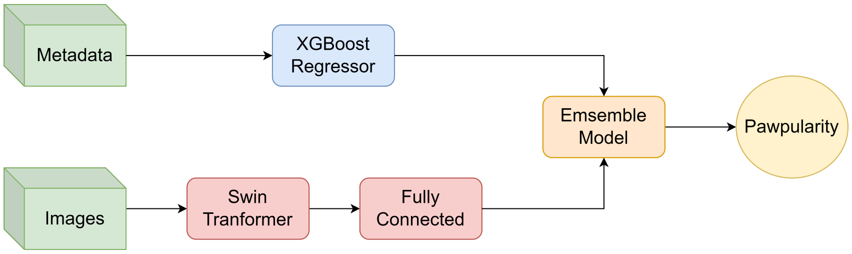
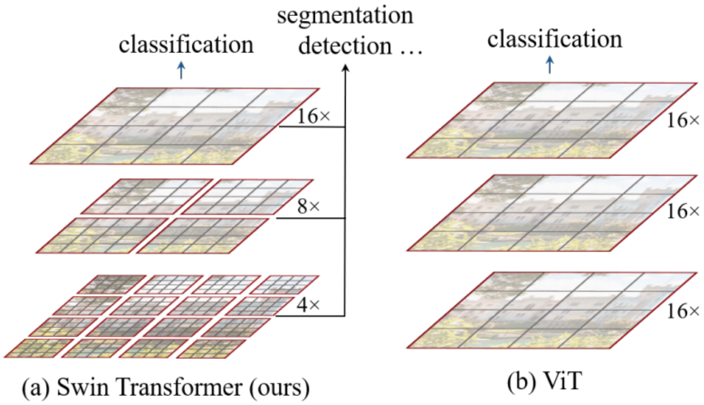


Figure 4: Final solution architecture.

As shown in the figure above, the Swin Transformer will be the backbone of our neural network, being responsible for feature extraction, which will be properly processed by the fully connected layer. In addition, we will also process the metadata features using an optimized distributed Gradient Boosting library for decision trees, XGBoost. Finally, after both models made their guesses, we use Optuna to maximize the respective Ensemble score on a validation set.

3.4.1 Swin Transformer

Fundamentally, our model will rely on a pretrained Swin Transformer as the backbone of our neural network, extracting valuable features which will be used by the fully connected layer. Over the pretrained model, we will train the model according the given dataset. Furthermore, we reframed the problem as a classification task instead of regression one, normalizing the target Pawpularity to [0, 1]. Therefore, we use Binary Cross Entropy Loss, which will better penalize worst predictions.



(a) Swin Transformer (ours)

(b) ViT

Figure 5: Swin and ViT approaches comparison.

The Swin Tranformer will calculate self-attention among images by 'shifting windows'. The shifted windowing approach was what brought greater efficiency when compared with other transformers (Figure 5). Finally, the complete Swin Transformer architecture is presented in Figure 6.

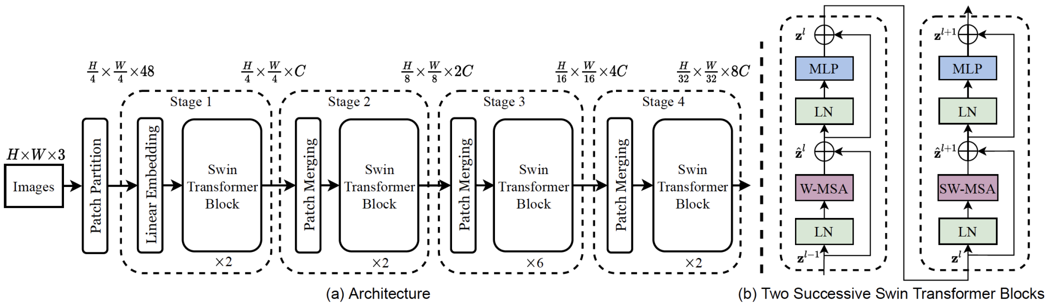


Figure 6: Swin architecture.

Moreover, a Stratified 10-Fold cross validation was implemented. In order to provide a uniform distribution, each strata will be fetched according percentiles. Finally, we will train the model for each fold, where the final Swin prediction will be the mean of every fold prediction.

3.4.2 XGBoost

XGBoost tries to improve single weak models (in this case, decision trees) by combining them together, in order to generate a collectively strong model. Moreover, the addition of the generated weak models is formalized as a Gradient Descent algorithm, over an objective function. Each of the decision trees is added one at a time, and trained using the residual errors of their predecessors as labels, therefore, at each iteration, the focus is directed to the samples which have not yet been accurately predicted. XGBoost has been known to achieve impressive results, being the most used Gradient Boosting implementation for tabular data, regarding Kaggle competition winning solutions. Consequently, we decided integrate this model and see how it would perform on the given metadata features.

Similarly to the Swin Transformer, a Stratified 10-Fold cross validation was implemented and the final predictions resulted from calculating the mean of the ones from every fold.

3.4.3 Ensemble Model using Optuna

At first, we relied on the Swin Transformer to make our predictions in the competition using the pictures, and on XGBoost to make our predictions using the metadata features, individually. However, we decided to build an Emsemble Model which would take into account the predictions from both individual implementations, because, even though the Swin Transformer had the best performance so far, perhaps XGBoost was also extracting useful signals from the metadata that would complement the ones of Swin. Accordingly, we used Equation 1, where we assign a weight to each model and calculate the corresponding weighted average for our predictions.

$$P = \sum_{K} w_k \cdot P_k, \quad \text{where} \sum_{K} w_k = 1 \tag{1}$$

We started by assigning weights according to the performance of each model during cross validation, so the Swin Transformer would have the largest weight. Afterwards, we decided to further tune the weights using Optuna, a fast AutoML framework which automates the trial-and-error procedure of optimizing hyper-parameters. Optuna uses a Bayesian Optimizer in order to intelligently focus the searching process in the ranges of values that present most potential. In the end, we were able to find an optimal combination and achieve improved results, which are presented in the following section.

4 Experiments

We have made a few submissions on *Kaggle* for each of the implemented models. Their respective score was calculated using the Root Mean Squared Error (RMSE) between our predictions and the correct labels. The best results for each model may be found in Table 1.

We have noticed in some of our testing that our model currently has a problem of high bias, resulting

in underfitting the data. As we investigated more about the problem and read public notebooks from the competition, we have concluded that the majority of the teams also is facing a similar situation. In fact, no mather how complex other team solutions could be, our best score is not that far from the best score in the competition, which is currently 17.50 (RMSE).
As shown on the scores table, both CNN and XGBoost models do not have outstanding results when compared to the baseline, where we predict the mean. Actually, these models will fail generalizing either to the train or test set. However, when the XGBoost model was ensembled with the pretrained Swin Transformer model, it slightly increased performance.

Model	**Public Score** (RMSE)
Baseline (Mean)	20.59
CNN K-Fold	20.56
XGBoost K-Fold	20.49
Swin Transformer K-Fold	18.04
Swin Transformer K-Fold + XGBoost K-Fold	**17.88**

Table 1: Different approaches and respective scores.

5 Conclusion

In conclusion, we started tackling this challenge as beginners in the field of Machine Learning. However, we were able to incrementally improve our solutions by performing thorough research on the topic and on the problem itself. Therefore, this project provided us an enriching learning experience, where we had the opportunity to apply the knowledge and techniques learned from the Machine Learning course, as well as from our own research, on a relevant real-work problem.
Regarding future work, we intend to further focus on the presented high bias situation, in order to conclude whether or not it is actually possible to extract more valuable signals from the given dataset, since it is extremely noisy.

6 Work Contributions

Both Armando and David started the project by implementing a first approach using a Convolutional Neural Network and K-Fold cross validation. Due to the poor scores of our initial model, we decided to tackle the different sides of the challenge in parallel, where Armando implemented a XGBoost model for the metadata, while David focused on integrating a Swin Tranformer model for the images, both using Stratified K-Fold with percentile distribution. Additionally, Armando designed an Ensemble model which would take into account both the Swin Transformer and the XGBoost predictions, using Optuna for optimal weight distribution. Both the elements worked together and cooperatively for the report.

References

[1] Kaggle and Petfinder.my. *Petfinder.my - Pawpularity Contest*. Sept. 2021. URL: `https://www.kaggle.com/c/petfinder-pawpularity-score`.

[2] PetFinder.my. *Cuteness meter*. URL: `https://www.petfinder.my/cutenessmeter`.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Vol. 25. 2012, pp. 1097–1105.

[4] Christian Szegedy et al. "Going Deeper With Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[5] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[6] Gao Huang et al. "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[7] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[8] Gao Huang et al. "CondenseNet: An Efficient DenseNet Using Learned Group Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[9] Mingxing Tan and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. 2019, pp. 6105–6114.

[10] Xiang Li et al. "Selective Kernel Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[11] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: `2010.11929 [cs.CV]`.

[12] Xizhou Zhu et al. *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. 2021. arXiv: `2010.04159 [cs.CV]`.

[13] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: `2103.14030 [cs.CV]`.

[14] Zihang Dai et al. "CoAtNet: Marrying Convolution and Attention for All Data Sizes". In: *CoRR* abs/2106.04803 (2021). arXiv: `2106.04803`. URL: `https://arxiv.org/abs/2106.04803`.

[15] Qizhe Xie et al. "Self-training with Noisy Student improves ImageNet classification". In: *CoRR* abs/1911.04252 (2019). arXiv: `1911.04252`. URL: `http://arxiv.org/abs/1911.04252`.

[16] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *ArXiv* abs/1905.11946 (2019).

[17] Hieu Pham et al. "Meta Pseudo Labels". In: *CoRR* abs/2003.10580 (2020). arXiv: `2003.10580`. URL: `https://arxiv.org/abs/2003.10580`.

[18] Aditya Khosla, Das Atish Sarma, and Raffay Hamid. "What makes an image popular?" In: *WWW* (2014).

[19] Aude Oliva and Antonio Torralba. "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope". In: *International Journal of Computer Vision* 42 (2004), pp. 145–175.

[20] Gygli M. et al. "The Interestingness of Images". In: *Computer Vision* (2013), pp. 1633–1640.

[21] Isola P. et al. "What makes an image memorable?" In: *Computer Vision and Pattern Recognition* (2011), pp. 145–152.

[22] Aditya Khosla et al. "Understanding and Predicting Image Memorability at a Large Scale". In: *ICCV* (2015).

[23] Sagnik Dhar, Vicente Ordonez, and Tamara L Berg. "High level describable attributes for predicting aesthetics and interestingness". In: *CVPR 2011*. 2011, pp. 1657–1664. DOI: `10.1109/CVPR.2011.5995467`.

[24] Jongpil Kim, Sejong Yoon, and Vladimir Pavlovic. "Relative spatial features for image memorability". In: *ACM Multimedia 2001* (2013), pp. 761–764.

[25] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: `10.1145/2939672.2939785`. URL: `http://doi.acm.org/10.1145/2939672.2939785`.