Petfinder Pawpularity Score prediction

Problem Statement

Millions of stray animals are on the streets without shelter around the world. To better adopt these animals, Pawpularity which is the visual appeal of the pet animals, significantly affects the adoption rate. Accurate predictions can help animal shelters and adoption platforms optimize photo selection, increasing adoption rate and reducing shelter overcrowding. This project analyzes the different algorithms, their strengths and weaknesses and proposes improvements.

Objectives

The purpose of this project is

- To review and understand existing approaches to predict Pawpularity using images as well as images with the meta data.
- To utilize the pretrained model with modification to improve performance for better prediction.

Data Description

The dataset consists of 9912 train images and 8 test images. Unlike other datasets, this dataset along with the images also contains meta which makes it different and needs different approaches to handle both images and meta data. The images are of different sizes and have three channels. The train and test files contain meta data for each image with label 0 and 1. The description of each feature is as below.

- **Focus** - Pet stands out against uncluttered background, not too close / far.
- **Eyes** - Both eyes are facing front or near-front, with at least 1 eye / pupil decently clear.
- **Face** - Decently clear face, facing front or near-front.
- **Near** - Single pet taking up significant portion of photo (roughly over 50% of photo width or height).
- **Action** - Pet in the middle of an action (e.g., jumping).
- **Accessory** - Accompanying physical or digital accessory / prop (i.e. toy, digital sticker), excluding collar and leash.
- **Group** - More than 1 pet in the photo.

- **Collage** - Digitally-retouched photo (i.e. with digital photo frame, combination of multiple photos).
- **Human** - Human in the photo.
- **Occlusion** - Specific undesirable objects blocking part of the pet (i.e. human, cage or fence). Note that not all blocking objects are considered occlusion.
- **Info** - Custom-added text or labels (i.e. pet name, description).
- **Blur** - Noticeably out of focus or noisy, especially for the pet's eyes and face. For Blur entries, "Eyes" column is always set to 0.

The final target variable is Pawpularity where the value is between 0 and 100. By looking at the Pawpularity distribution in Figure 1, most pets have low-to-moderate Pawpularity (20–40), only a few are highly popular, and the distribution is right-skewed with a spike at 100.

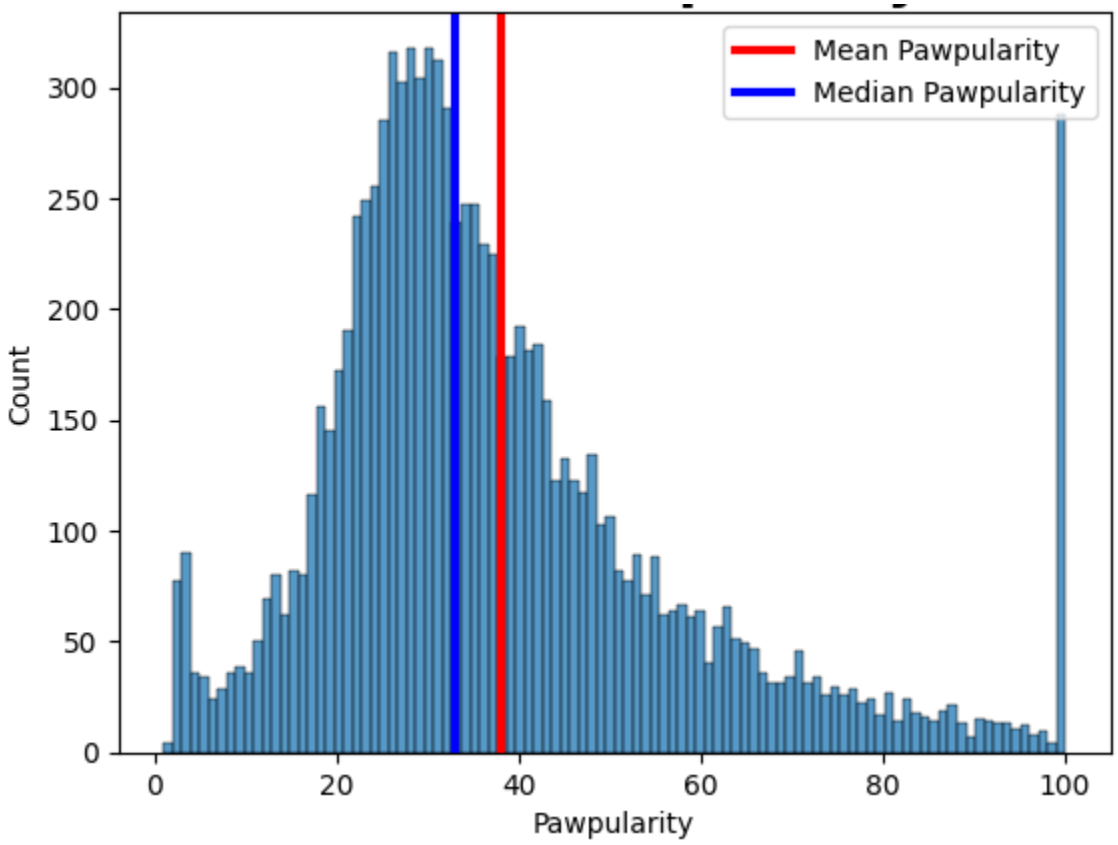

Figure 1 Pawpularity Score Distribution

By looking at each feature correlation **Figure 2**, the blur has minor correlation, However overall, there is not any significant correlation of all features with the target Pawpularity Score.

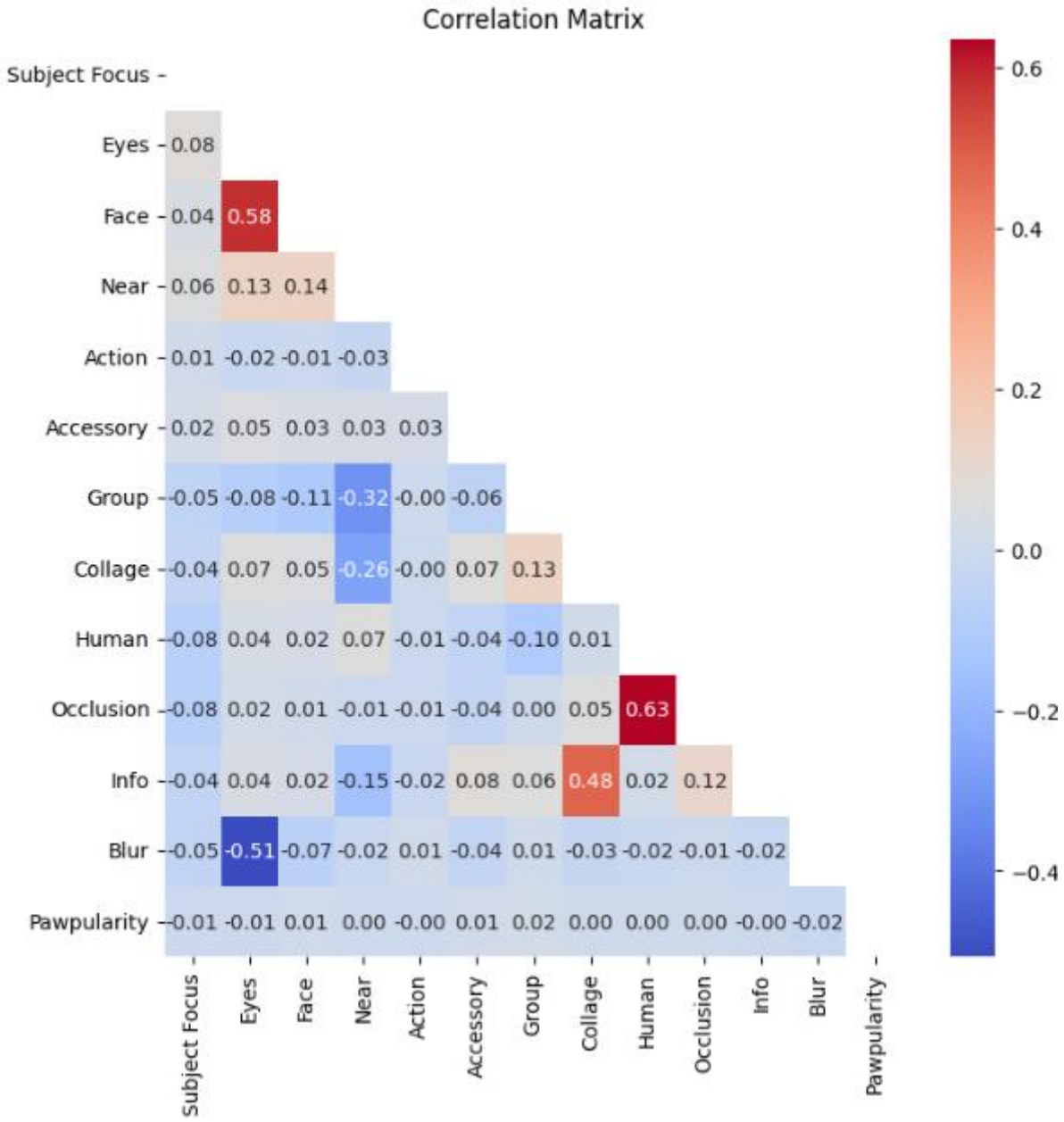

Figure 2 Correlation Matrix

But if we look at the features correlation with each other, then there is some correlation between Face and Eyes, Human and Occlusion. To better understand and analyze, we use the variance inflation factor (VIF) which determines the relationship with all other features. Hence from **Figure 3,** we can see the Face and Eyes have high correlation, and it needs to be handled by Regularization (early stopping, weight decay) while training the model.

	feature	VIF
2	Face	13.715668
1	Eyes	10.118170
3	Near	5.762924
9	Occlusion	2.073562
8	Human	2.064939
11	Blur	1.595109
7	Collage	1.452023
10	Info	1.412621
6	Group	1.163850
5	Accessory	1.090942
0	Subject Focus	1.048292
4	Action	1.010174

Figure 3 VIF Score

Related work

The two baseline codes are picked for comparison based on the images only as well as images with metadata consideration. They used different approaches to improve their results including different augmentation, cross validation etc. The structure is analyzed based on validation RMSE, due to no label/target for test data.

1. Vision Transformer with Metadata and Images

The first approach uses the Vision Transformer (ViT) architecture [1], which applies the Transformer model directly to sequences of image patches rather than relying on convolutional operations. This method has demonstrated competitive performance dependencies through self-attention mechanisms. In this implementation, a pretrained ViT backbone from the timm library (PyTorch Image Models) was used, and the extracted image embeddings were concatenated with tabular metadata features before passing through a fully connected network for final prediction. The model was trained using PyTorch, incorporating augmentation, dropout regularization, and a custom learning rate schedule to improve generalization.

It combines image embeddings from a ViT with metadata features. Using five fold cross-validation improves reliability by evaluating the model on five different validation sets. Although the average RMSE (17.80) is slightly higher compared to Swin Transformer, it reflects consistent performance across folds.

2. Swin Transformer with Images Only

The second approach uses the Swin Transformer [2] , which introduces a hierarchical architecture with shifted windows to compute self-attention locally. Unlike the first approach, they used images only without metadata. This approach prioritizes simplicity and computational efficiency, relying on the model's ability to capture rich visual features without auxiliary metadata

The implementation utilized the fastai library, which provides high-level abstractions for efficient training and hyperparameter tuning. The training pipeline included data augmentation, and a single validation split rather than k-fold cross-validation.

The summarized table for both approaches

	Approach1: (Image + Metadata)	**Approach 2: Image only**
Data	Image and metadata	Image only
Model	Vision transformer	Swin Transformer
Cross validation	Five fold	One fold
Strength	Considers extra metadata for better context and robust CV	Good for capturing both local and global features due to attention and shifted window
Weakness	More complex	Ignore metadata, may underperform and increase risks of overfitting due to single fold
Validation RMSE	17.88	17.55

References

[1] Manav, "Transformers Classifier Method Starter Train," Kaggle, Petfinder.my - Pawpularity Contest,2021.[Online].Available: https://www.kaggle.com/code/manabendrarout/transformers-classifier-method-starter-train/notebook

[2] Tanlikesmath, "PetFinder Pawpularity EDA + FastAI Starter," Kaggle, Petfinder.my - Pawpularity Contest, 2021. [Online]. Available: https://www.kaggle.com/code/tanlikesmath/petfinder-pawpularity-eda-fastai-starter/notebook