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 PREPARATION AND ANALYSIS

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.

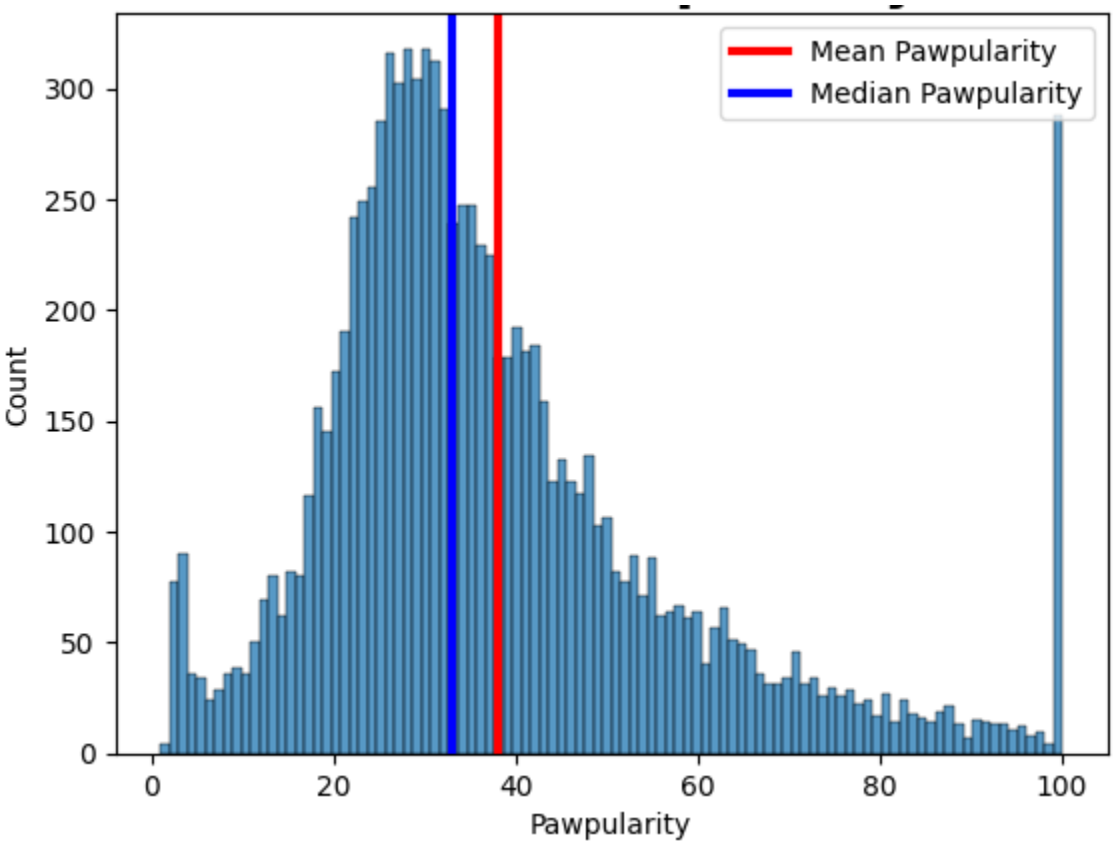
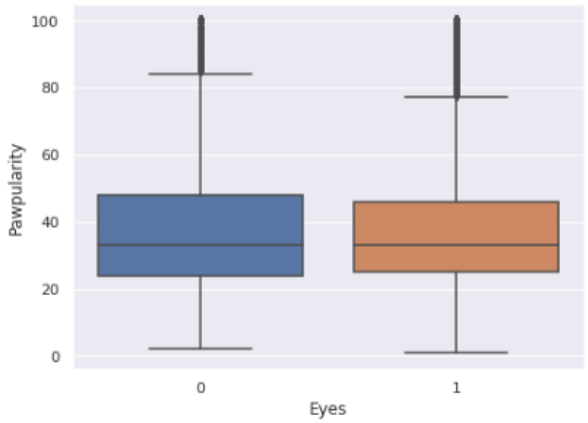
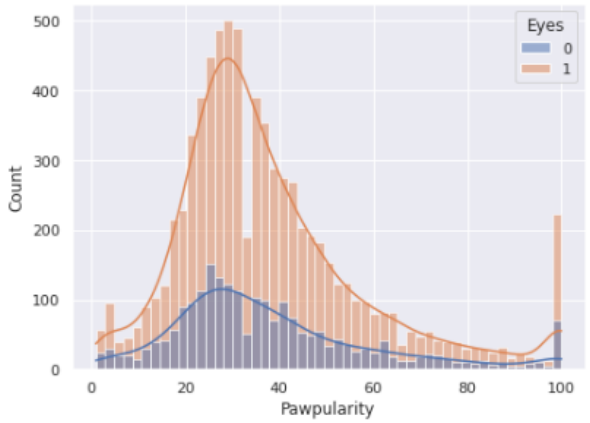


Figure 1 Pawpularity Score Distribution

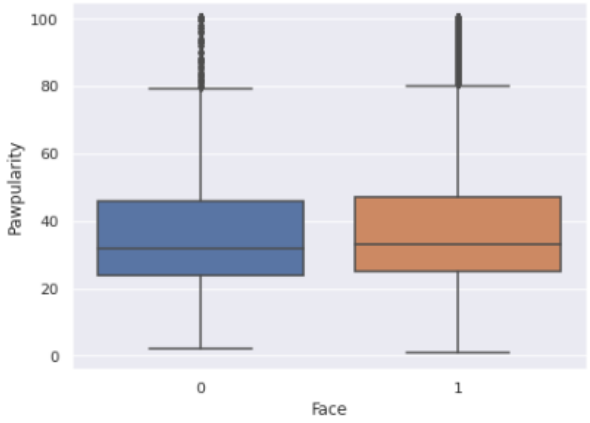
Likewise by looking at the box plots of each input variable, it has very similar ranges (mean, median, distribution shape). which means having the feature vs not having the feature do not meaningfully change the Pawpularity(target) score. Also the data is imbalances, the number of records having 0 for some feature are more while for other number of records having 1 are more.

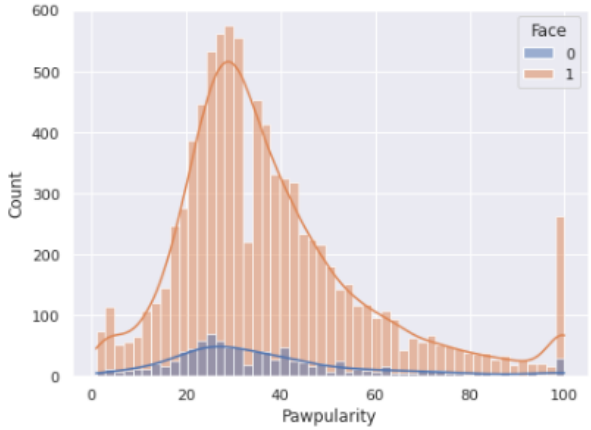
Eyes



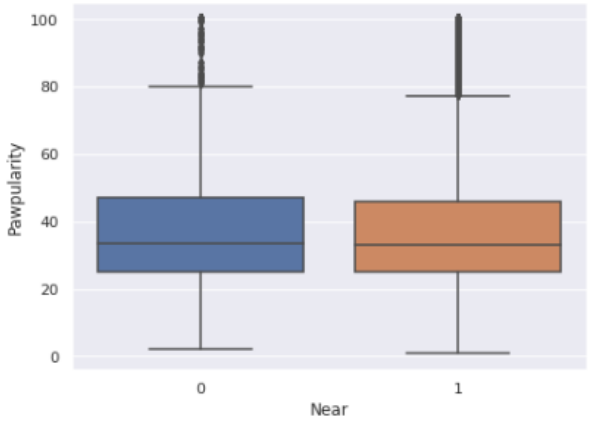


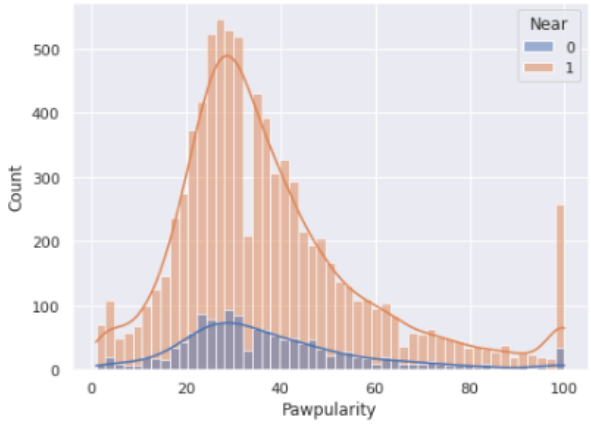
Face





Near





Considering this, the metamodel will not be able to perform well, and we need to good for the image models.

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

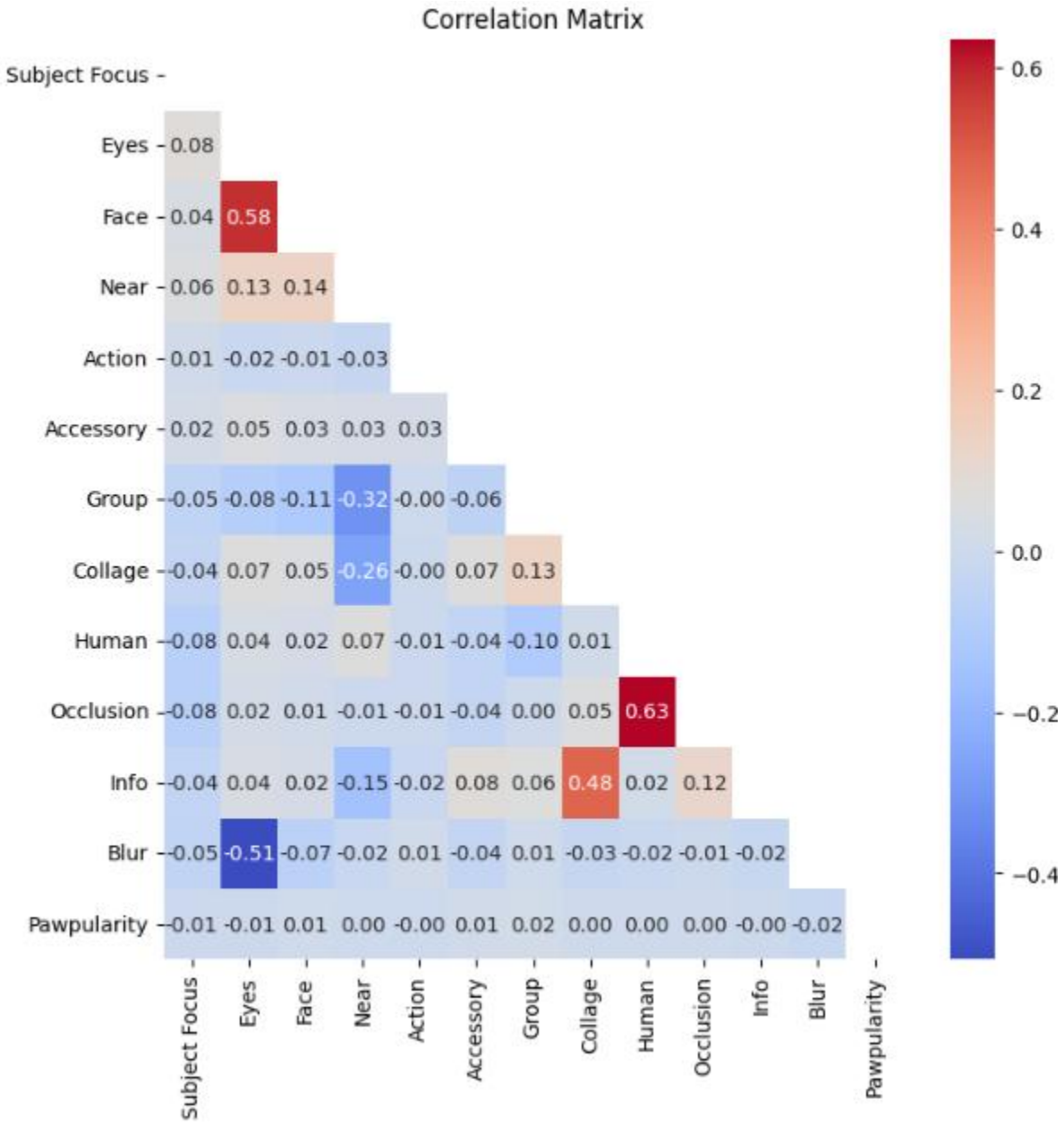


Figure 2 Correlation Matrix

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 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 unavailability of target values.

Vision Transformer with Metadata and Images

The Vision Transformer (ViT) is a deep learning model that applies the Transformer architecture, originally designed for natural language processing, to image understanding. Instead of using convolutional filters like CNNs, ViT treats an image as a sequence of small image patches, similar to how words form a sentence. The model learns relationships between all patches using self-attention allowing it to understand both local details and global context across the image. The image is divided into patches, learning how different parts of the picture relate to one another, and combining this information to predict outcomes, the pawpularity score of a pet photo.

To find the Pawpularity value, ***(Manav 2021)*** used the Vision Transformer (ViT) architecture in Petfinder Pawpularity Contest. 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. The final layer of VIT is replaced and connected to 128 nodes in the next layer followed by 64 nodes. By using five

fold cross validation, the model achieved the average validation RMSE 17.88. This approach has the good point that it considers the image as well as meta data which will help model to get better results and generally looks at all parts of the image together and captures the overall scene well. However, as VIT processes all image parts together at the same time that can lead to slow processing.

Swin Transformer with Images Only

Swin Transformer introduces a hierarchical architecture with shifted windows to compute self-attention locally. This model divides each 224×224 image into 4×4 patches. It is designed to efficiently model both local and global visual features. Instead of analyzing the whole image at once (like ViT), Swin divides the image into small non-overlapping windows and applies self-attention only within each window. Swin transformer is like a Transformer-based CNN. Unlike the first approach, ***(Tanlikesmath 2021)*** 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 and achieved the validation RMSE 17.55. The validation RMSE is lower, but it ignored metadata and considered single fold, which may underperform and increase risks of overfitting due to single fold.

Swin Transformer with Metadata and Images

(Wang, Y., & Liu 2022) proposed a Swin transformer and MLP architecture (PETS-SWINF) that combined the image features and image metadata to balance image and metadata contributions for predicting a pet photo's "Pawpularity." The authors argued that many prior solutions focused primarily on image-only models and did not effectively exploit metadata; PETS-SWINF explicitly combined them with image representations, improving validation RMSE slightly compared to the image-only variant. PETS-SWINF therefore represented a practical example of multimodal fusion (vision + tabular metadata) for image regression tasks and demonstrated that careful metadata modeling and fusion can produce better results over image-only baselines. Meta data and Images were trained separately and then their validations were combined to produce the final predictions. Additionally, weights were assigned to the model depending on the validation RMSE, the higher RMSE model is penalized more, and the influence of that model is reduced. For example, if the meta model prediction is equal to mean of the target, then their weights will be 0 and there will be no effect of model and only the image model will give the final predictions. The PETS-SWINF approach demonstrates notable strengths by integrating image features with metadata, enabling a richer representation of factors influencing pawpularity. This fusion, combined with an adaptive weighting mechanism, allows the model to balance contributions from visual and tabular data effectively.

METHOD

We utilized the PetFinder.my dataset from the 2021 Kaggle Competition, which consists of pet images and associated metadata, including features such as 'Subject Focus,' 'Eyes,' and 'Face,' to predict the Pawpularity score. The dataset was split into four folds for cross-validation, ensuring robust evaluation. We retained the metadata model architecture as described in PETS-SWINF, a fully connected neural network processing 12 metadata features where each input metadata vector is projected through the 50-dimensional GloVe-based embeddings into a lower-dimensional vectors. Images were preprocessed using standard augmentations (e.g., resizing, normalization) for the Swin Transformer model, with an input resolution of 384x384, an increase from the 224x224 resolution used in PETS-SWINF. Batch size of 16 with the 10 epoch for metadata model and 5 epochs for Image Model are configured. For the image model, we upgraded to a Swin Transformer Large (Swin-L) with an input resolution of 384x384, compared to the Swin Transformer with 224x224 resolution in PETS-SWINF. The larger input resolution allows the model to capture finer visual details. The combined RMSE is calculated based on the metadata prediction and image model prediction.

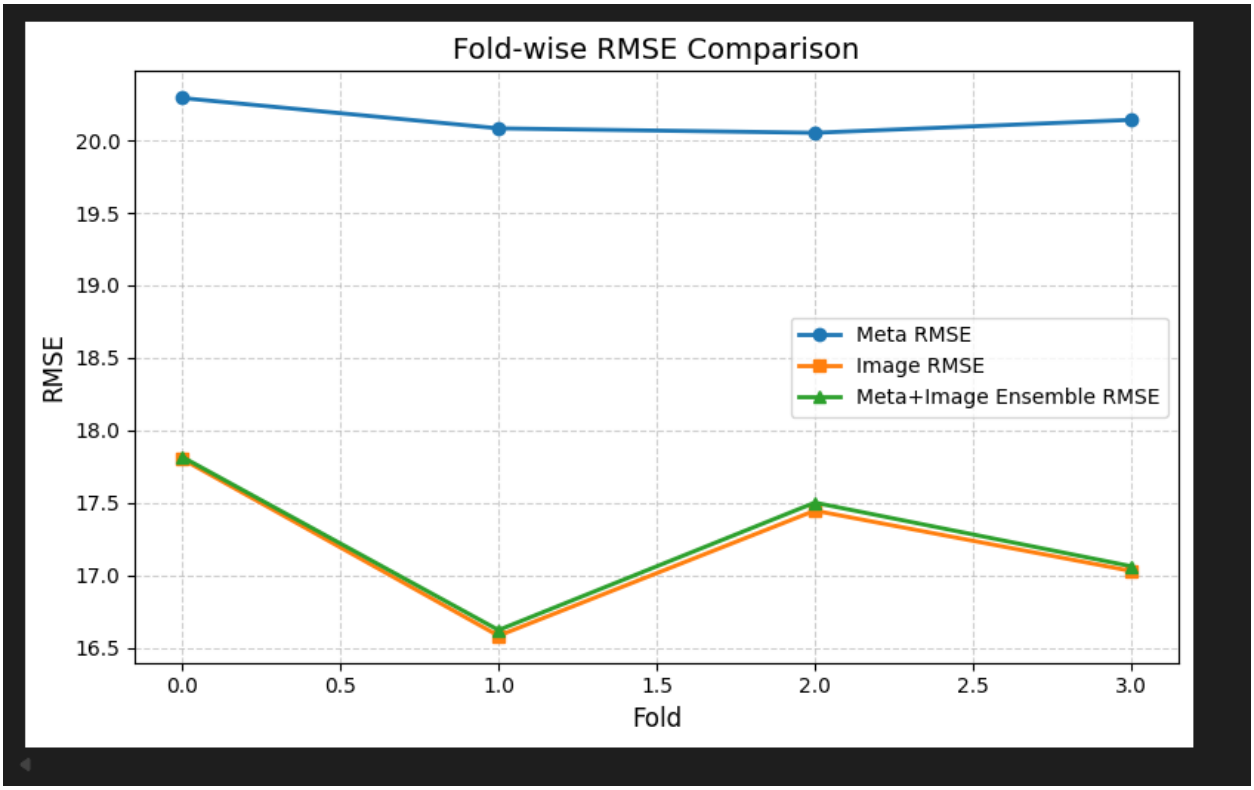
Experimental Setup

The hardware and software used in the project are:

Name	Specification
Operating System	Window 11
Python	3.13.7
GPU	NVIDIA GeForce RTX 3060, 12 GB
Deep Learning Framework	Pytorch 2.8.0+cu128
Cuda and cuDNN version	Cuda 12.8, cuDNN 91002
RAM	32 GB
Processor	AMD Ryzen 7 PRO 7745 8-Core Processor (3.80 GHz)

RESULTS

The image-only model achieved a lower validation RMSE compared to the metadata model and to the ensemble computed in this project. The ensemble RMSE calculated using the proposed weighting method did not outperform the image model. However, the image model alone provided strong performance, and its RMSE was slightly better than the ensemble RMSE reported in the PETS-SWINF paper, despite using fewer number of folds. As it can be seen in the graph below, image model performed well across all the folds.



the Metamodel did not perform well across all the folds which means if we include this metamodel to our image model by the fusion formula. it will increase the RMSE. Hence the model was trained on just 5 epochs, so there is fluctuation of RMSE fold wise in the image model, increasing the epochs can reduce the fluctuation.

$$meta_{weight} = (max(0, 20.5 - RMSE_{meta}))^2 \quad (1)$$

$$image_{weight} = \left(max\left(0, 20.5 - RMSE_{image}\right)\right)^2 \quad (2)$$

$$RMSE_{ensemble} = RMSE_{meta} * meta_{weight} + RMSE_{image} * image_{weight} \quad (3)$$

Where 20.5 is the standard deviation of target pawpularity which means if the model predicts the target equal to the mean, then std will become equal to RMSE, and the model will not contribute to the prediction, RMSEmeta is the RMSE of metamodel and likewise RMSEimage is the RMSE for image model(swin transformer).

The fold wise RMSE is shown below

```
Fold-wise RMSE & Weights Table:

 Fold  Meta RMSE  Image RMSE  Ensemble RMSE  Meta Weight  Image Weight
    0    20.2918     17.8018        17.8166       0.0059        0.9941
    1    20.0827     16.5832        16.6225       0.0112        0.9888
    2    20.0519     17.4465        17.5014       0.0211        0.9789
    3    20.1411     17.0300        17.0629       0.0106        0.9894

Overall Statistics (Across Folds):

   Model  Mean RMSE  Variance  Std Dev
    Meta    20.1418    0.0085   0.0923
   Image    17.2154    0.2078   0.4559
Ensemble    17.2508    0.2032   0.4508
```

The reason metamodel is not performing well due to each input variable has very similar ranges (mean, median, Quartile). which means having the feature vs not having the feature does not influence the target score. Also, the data is imbalanced, the number of records having 0 for some features are more while for other number of records having 1 are more. From the target distribution, the majority of samples are mid-range (20–50), while very few are near 0 or 100. Models trained on this will tend to predict mid-range values and struggle with rare high scores. Further, the Metamodel did not perform well across all the folds which means inclusion metamodel to our image model by the fusion formula can add noise and affect the performance.

However, compare to the existing models discussed, the proposed model with just images outperforms the others.

Table 1 RMSE Comparison

	Vision Transformer VIT (Metadata+Images)	Swin Transformer (Images only)	Swin Transformer PETS-SWINF (Metadata + Images)	Proposed Extended Swin Transformer(Images)
Validation RMSE	17.88	17.55	17.41	17.21

CONCLUSION AND FUTURE DIRECTION.

This experiment demonstrates that a single image model with higher input resolution (384×384) can achieve competitive pawpularity prediction performance. The Swin Transformer architecture effectively captures visual patterns relevant to pet popularity. While we implemented the ensemble approach from the original paper, we found the only image model provided the most reliable results.

In the future, multiple transformer models combination and addition of folds as well as epochs can result in better performance.

References

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

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

Wang, Y., & Liu, Y. (2022). PETS-SWINF: A regression method that considers images with metadata based Neural Network for pawpularity prediction on 2021 Kaggle Competition" PetFinder. my". *arXiv preprint arXiv:2201.06061*.