# Task 3 - LIME

The goal of this task is to derive an explanation for image classification using Local Interpretable Model-agnostic Explanations (LIME). Given for the task are an image classifier (ResNet 50 trained on ImageNet) and 10 images of which the classification should be explained.

LIME works by reducing the number of features to a human comprehensible number and exploring which of these is most important to get to the given classification. This simplification is viable because LIME only explores the decision boundaries locally around the given data point and doesn't try to explain the entire decision process of the model.

For image classification this is done by splitting the input image into a small number of segments. The classifier is then run on only subsets of these segments while the remaining segments are masked out. A simple model is trained on what segments were used and the classifiers confidence for the correct class of the image. By this the model learns which segments are most important for the classifier to make the correct decision and also which segments contradict the classifiers decision. These segments can be highlighted in the output image to explain what parts of an image were relevant for the decision of the classifier.

For this task LIME was performed according to this tutorial: LIME-Tutorial. The code that was used is found in *task3/lime.ipynb*.

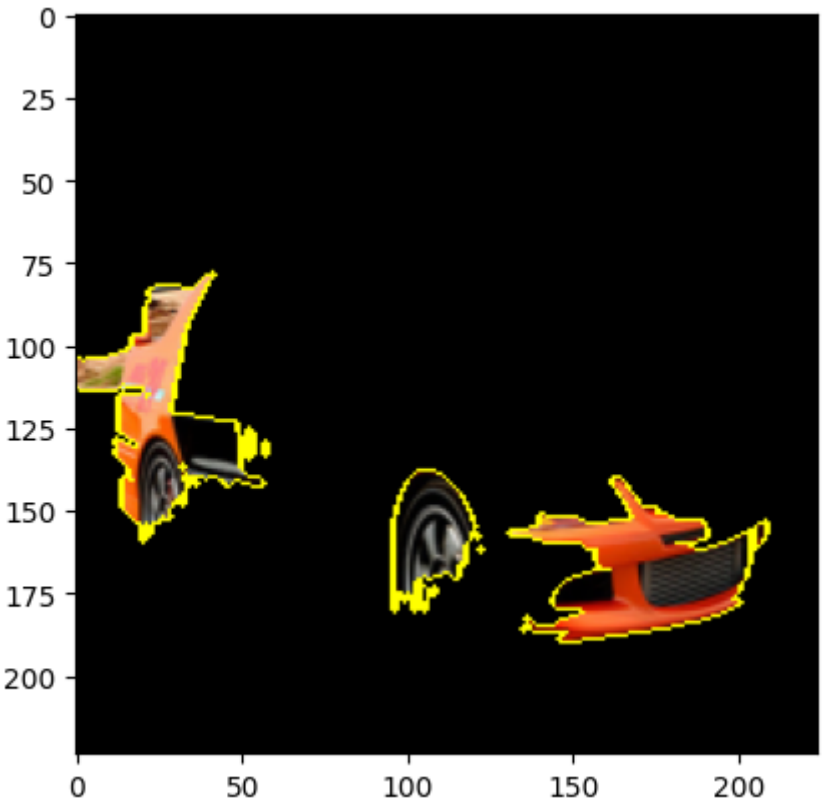
To create each of the explanation an explain instance had to be configured. Such an explain instance takes in a multitude of parameters that were fine tuned for each image to give the best possible result in as short of a time as possible. These parameters were submitted to the assignment API for grading of the explanation. Unfortunately the parameters had to be submitted as pickle-serialization, which does not support the serialization of lambda functions. This made it impossible to use anything but the default for the parameters *segmentation_fn* and *model_regressor* which expect wrapper functions to function of the *skimage* and *sklearn* modules.
During initial testing very good results were achieved using `slic(img, n_segments=x, compactness=y)` for segmentation which allows to specify the number of segments and how they are selected based on color and position.

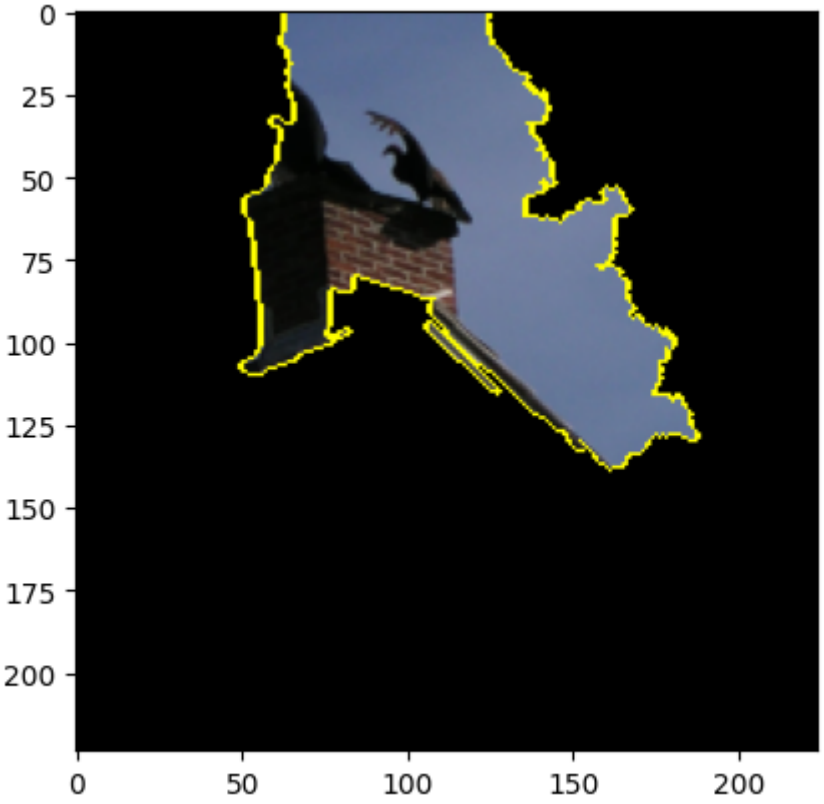
This leaves the following parameters for tuning of explain instance:

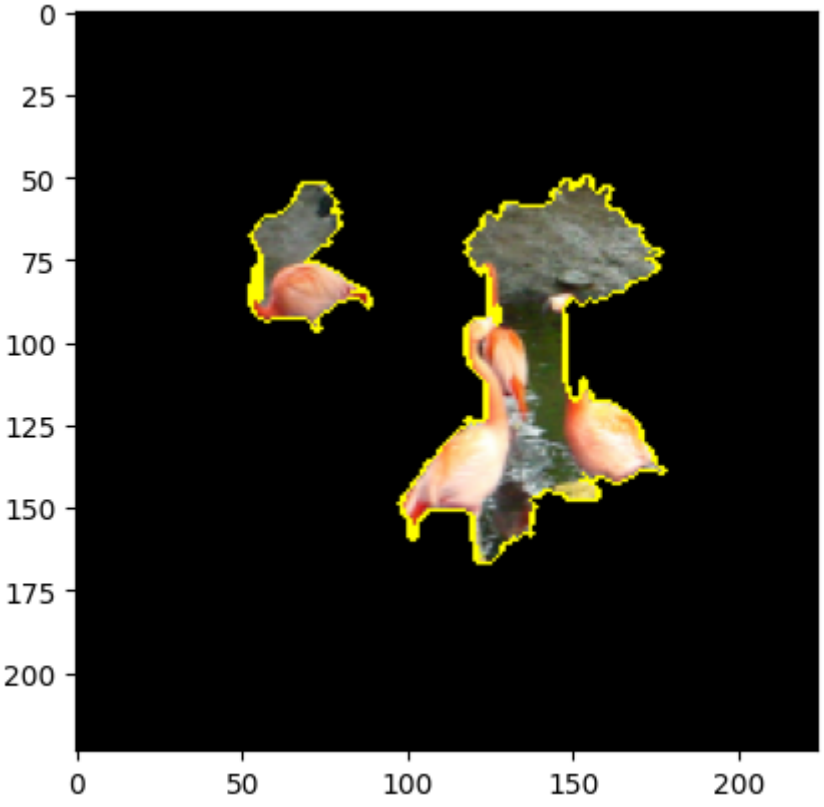
- labels/ top_labels: labels that the models tries to explain. Set to top_label = 1 to explain only the label the classifier is most confident in. Exception is the image labeled as "kite" were it was set to 3 since the models classification is wrong.
- hide_color: color used to map out the unused segments. Set to either white, grey, black or None (defaults to average color of image).

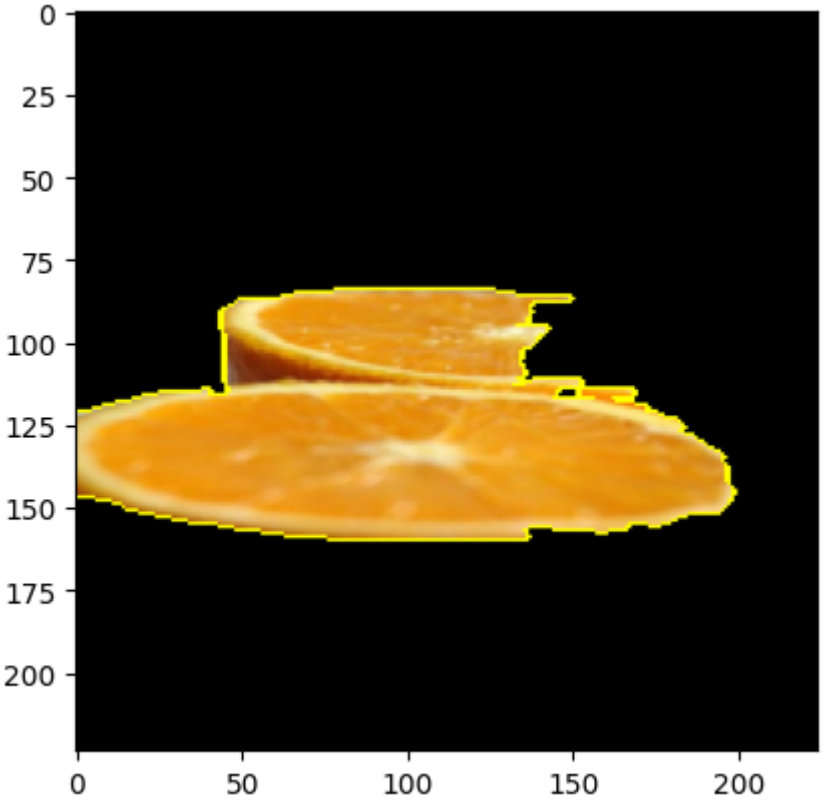
- num_features: maximal number of features the explanation model should have. Corresponds to number of segments in the image. Lower number of features reduced complexity and are easier to comprehend. Number is set between 4 and 8, since bigger numbers usually didn't result in a more detailed/ more complete explanation mappings.
- num_samples: size of the neighborhood created to train the explanation model. Set as low as possible while still producing a good mapping to achieve a short execution time. Usually set around $2^{num\_features}$ since this should include all possible subsets.

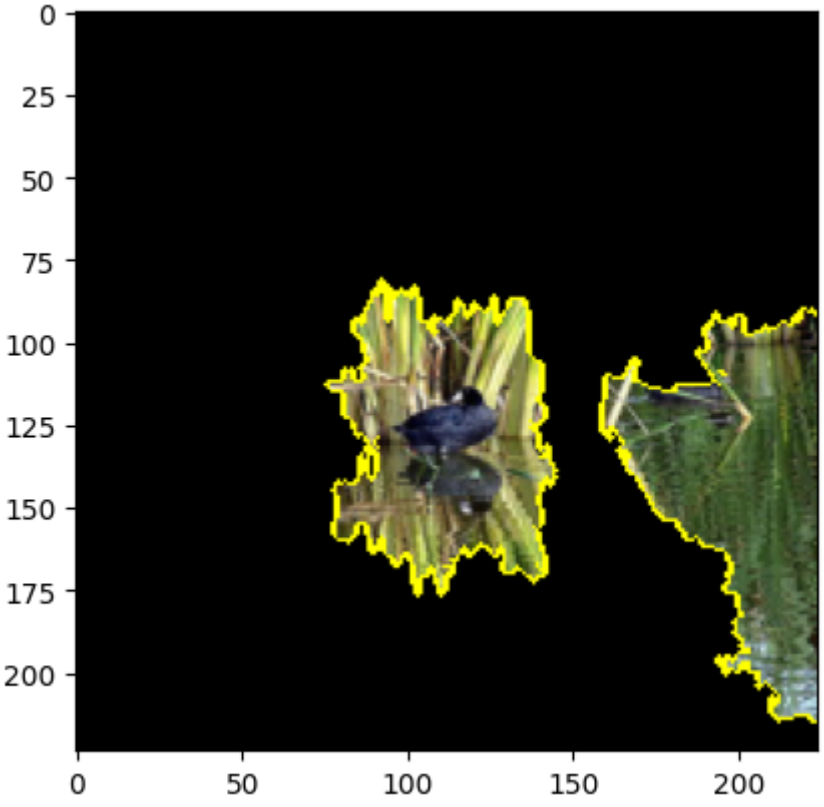
The following images show the areas that were deducted to be the most relevant to the classification, given the mentioned configurations and limitations, for each of the sample images:

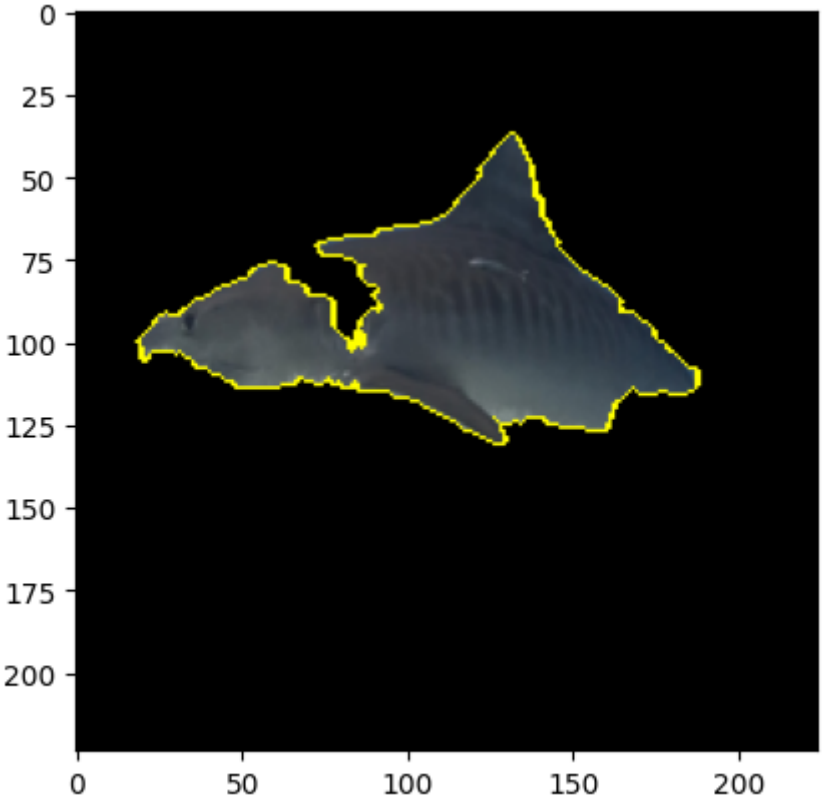


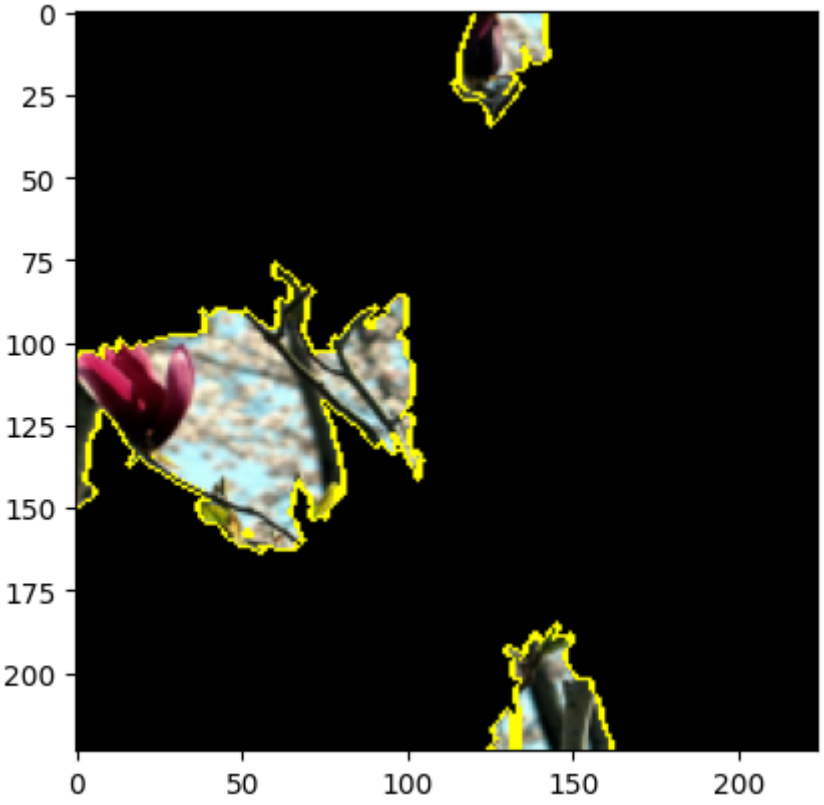

0
25
50
75
100
125
150
175
200
0
50
100
150
200

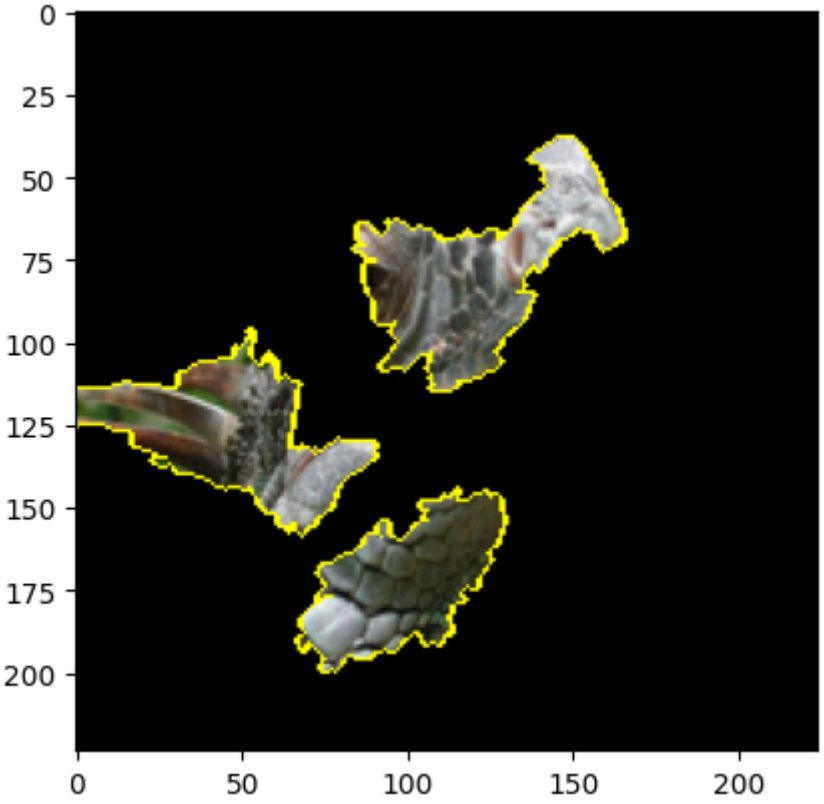

0
25
50
75
100
125
150
175
200
0
50
100
150
200

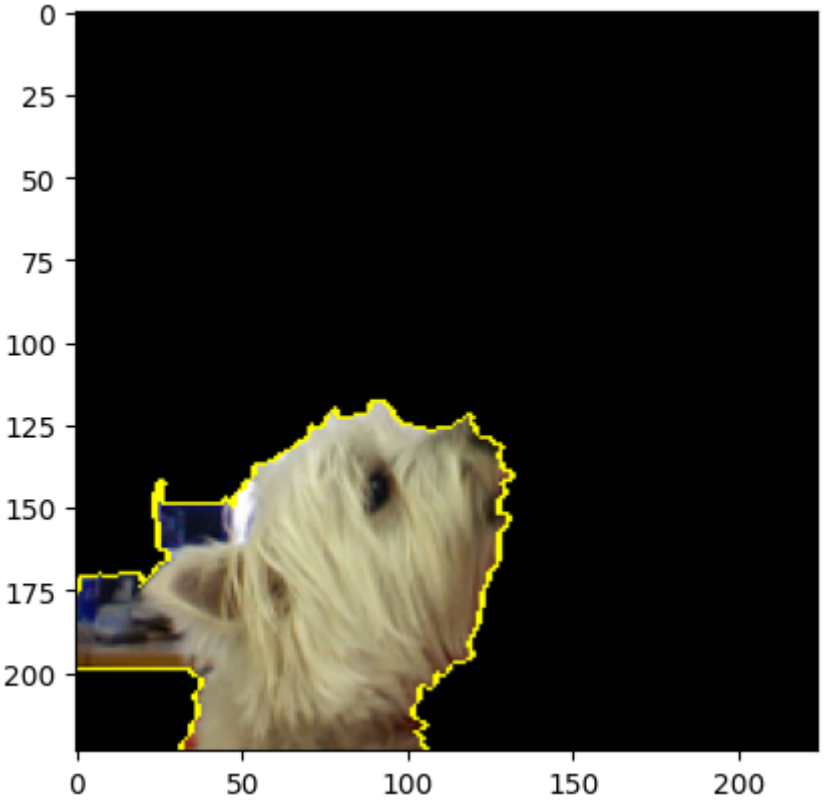

0
25
50
75
100
125
150
175
200
0
50
100
150
200

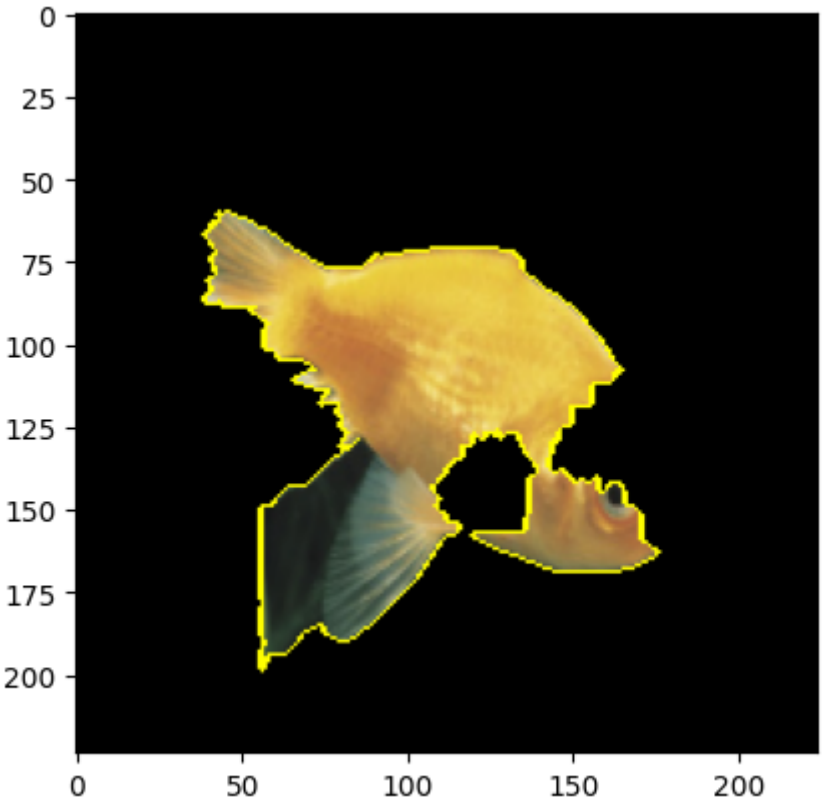

0
25
50
75
100
125
150
175
200
0
50
100
150
200


0
25
50
75
100
125
150
175
200
0
50
100
150
200


0
25
50
75
100
125
150
175
200
0
50
100
150
200


0
25
50
75
100
125
150
175
200
0
50
100
150
200


0
25
50
75
100
125
150
175
200
0
50
100
150
200




These LIME annotations were graded by the assignment API as foloows:
`{'avg_iou': 0.334783209795884, 'avg_time': 1.9471065044403075}`

## Conclusion

The highlighted regions in the cropped images correspond to the areas that most influenced the classifier's decision. In most examples, these regions include the most distinctive features of the depicted object. This suggests that LIME was effective in explaining the model's behavior.

For simpler objects—such as the orange or the goldfish—the model's decision appears to rely on the entire object, likely due to its shape and color. In contrast, for more complex objects like the racecar, the classification depends on several finer details scattered across the image, including the wheels, grille, and spoiler.

Interestingly, for more challenging classifications, the model also seems to factor in background elements. For example, in identifying a vulture, features like the roof and sky contributed to the decision, while for the coot, the surrounding water played a role.