We used a convolutional neural network, specifically the Keras Xception model, we retrained the model. We split our data into training and test sets using a 90-10 split. We retrained the Xception model, using our 90% training set, which resulted in a much higher prediction accuracy than when we tried to use transfer learning. We applied several transformations to the training images, including width_shift, height_shift, rotation, zoom, and vertical_flip. The values used for these transformations were initially determined by thinking of which transformations could result in an image that is different, but that still resembles a valid image that could be taken of these buildings. An example of this is in the vertical_flip transformation. We reasoned that a vertically flipped image would produce a valid building image, but that a horizontally flipped image would not be helpful, because we will not be testing with any upside-down building images.

We estimated our future performance based on how well our model predicted labels for the 10% validation set. We calculated the recall, precision, and accuracy for each building label by interpreting the predictions as one vs rest, so that we can count the total number of False Positives and True Positives, and use these with the total number of images for each building to calculate the above metrics for each. Using these calculated values, we can make an estimate for which will have the highest/lowest values of these in the future. From our model, we came up with the following predictions:

Building with worst recall: AA
Building with best precision: FH
Overall accuracy: 0.9696

We determined which buildings did the best and worst overall by making a precision recall curve and plotting each building to see which were closest and furthest from the optimal spot in the top right corner. The buildings that did the best were DT, RO, and LB. This is likely because these buildings all have a distinct pattern which looks the same from every angle of the building. The buildings that our model did the worst on were CB, MC, AA, and CT. This is likely because these buildings are either easily mistaken for other buildings (CB and CT), or because they look very different from different angles (MC and AA), which may confuse the model. More specifically, CB and MC had high recall, but low precision, which means that it had a lot of True and False Positives. This indicates that other buildings are very often mistaken for CB or MC. CT and AA had high precision, and low recall, which means that it had few True and False Positives. This indicates that the model rarely thought that any building was CT or AA, whether it was or not. These best and worst buildings highlight the strengths and limitations of our model. The strengths are its ability to identify buildings with consistent exteriors, and the limitations are its inability to identify buildings with very similar appearances as another building, or with inconsistent exteriors.

If we had more time, we would have done more tests to find the optimal values for the preprocessing transformations of training images. It is possible that the training would have benefitted from some of the transformations being more/less drastic, but in our testing we did not have time to try changing one at a time and retraining the model to see the difference that each individual change makes. With more time we could also try training using more methods of optimization to see if any are better suited to this task.