HouseClassifier.com

dn-ds

What is HouseClassifier?

A web app that classifies a given image of a house into one of six types.

Six architectural styles considered:

Cape Cod   **Mediterranean**

Colonial   **Ranch**

Craftsman   **Tudor**

Cape Cod

Central chimney, flat front facade, moderately-pitched side-gabled roof, often dormer windows.

Colonial

Symmetrical design, rectangular shape, central front door, side-gabled or hipped roof.

Types of Houses (continued)

Craftsman

Deeply overhanging eaves, often front-gabled roof, front porch.
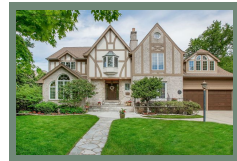

Mediterranean

Low-pitched roof, red clay roof tiles, stucco walls, arches.

Ranch

Single-story, often low-pitched long roofline, overhanging eaves.

Tudor

High-pitched roof, ornamental wood framing, large chimney.

Project Outline

Gathering and Processing the Images

⇓

Building and Training a Basic Convolutional Neural Network

⇓

Transfer Learning

⇓

Web App

Gathering and Processing the Images

- House images are downloaded using Google search and the Chrome extension *Download All Images*.
- Each image is manually inspected; mislabeled and fake images are deleted.
- Additional images are downloaded from *zillow.com*.
- 2,100 total images, 350 images per class.
- Duplicate images are removed.
- Training, validation, and test sets are prepared.
- The dataset is balanced, so *accuracy* is an appropriate evaluation metric.

Building and Training a Basic CNN

- Data augmentation layer.
- *He* initialization to mitigate the vanishing/exploding gradient problem that may occur at the beginning of training.
- Batch Normalization layer to mitigate the vanishing/exploding gradient problem that can return during training.
- Pooling layer to reduce the computational load and the number of parameters by downsampling inputs. The latter helps limit the risk of overfitting.
- Dropout layer to further mitigate overfitting.
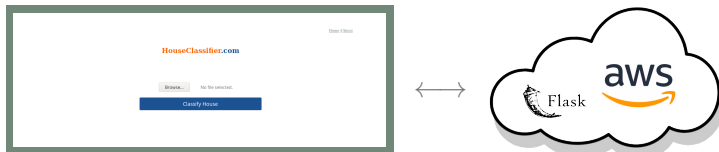- Accuracy score 0.33 on the test set.

- Idea: Use a model developed for one task as the base for a model on a second task.
- A pretrained Xception model is used as as a base model.
  - Developed for the *ImageNet Large Scale Visual Recognition Challenge*.
  - Trained on a dataset consisting of more than a million images.
  - Classifies a input image into one of 1000 diverse categories (e.g., sea lion, tractor).

- New classifier on top of the base model.
- The base model is customized in two ways:
  - Feature Extraction: Representations learned by the base model is used to extract meaningful features from the house images.
  - Fine-Tuning: Some of the top layers of the pretrained base model are unfreezed, and jointly trained along with a new classifier.
- Dropout layer to mitigate overfitting, which can be severe especially during fine-tuning
- Final model accuracy score of 0.93 on the test set.
- The model was trained on an AWS EC2 c5a.9xlarge instance.

- When an image is submitted, it is accepted and sent to the model using jQuery and PHP
- The model then processes the image and returns a prediction.
- The model is deployed on an AWS EC2 t3a.small instance using the Flask framework.

Main Tools and Packages Used