MSc Data Science Project Report

Improving House Price Estimation with Artificial Neural Networks and Satellite Imagery

Neil Scrivener
Student Number: 13147162
Supervisor: Vladislav Ryzhikov

Department of Computer Science and Information Systems

Birkbeck, University of London

September 2020

Academic declaration

This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I give my permission for it to be submitted to the JISC Plagiarism Detection Service. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Neil Scrivener

Abstract

Residential property is a pivotal element of the UK economy and informs monetary policy decisions. Emerging competition in the PropTech (Property Technology) sector is increasing the need for automated valuation models to be accurate. This project assesses whether visual feature extraction from satellite imagery can be used to improve house price estimations in a neural network model. Multiple datasets of property and localised attributes were combined and a baseline level of accuracy was established without visual features using ordinary least squares regression, random forest and artificial neural network models. Tabular data was combined with features extracted using a purpose-built convolutional neural network and ResNet50 and a new set of models were produced. A small reduction in mean absolute error from 0.132 to 0.127 was achieved with the visual features extracted from satellite images using the purpose-built convolutional neural network but due to difficulties with image overlap it is not clear whether the model would generalise to other areas of the country. The models using the ResNet50 features showed exceptionally poor performance for which no objective explanation could be determined.

Acknowledgements

I would like to thank my supervisor, Vladislav Ryzhikov for his calmness and patience throughout this project and the entire staff of the Department of Computer Science and Information Systems at Birkbeck for all their teaching and support throughout the last two years that has enabled me to complete this postgraduate degree. I'd also like to thank the members of the Birkbeck Brethren without whom I would have almost certainly had a breakdown, given up, died or a combination thereof.

Contents

1 Introduction **7**

2 Background **8**

2.1 The Hedonic Pricing Method 8

2.2 Feed Forward Networks 8

2.3 Convolutional Neural Networks 10

2.3.1 Feature Extraction 11

2.3.2 Transfer learning 12

3 Methodology **12**

3.1 Data gathering cleaning and preparation 12

3.1.1 Data gathering 12

3.1.2 Data merging 14

3.1.3 Data exploration 15

3.1.4 Data cleaning and preparation 17

3.1.5 Data preprocessing 20

3.2 Feature selection 21

3.3 Baseline models 23

3.3.1 Linear regression model 23

3.3.2 Random Forest 23

3.3.3 Baseline neural network model 24

3.4 Visual features 25

3.4.1 Data preparation 25

3.4.2 Model selection 27

3.4.3 Feature extraction 29

3.5 Combined feature models 29

4 Software and packages **30**

5 Evaluation **31**

5.1 Model Accuracy 32

5.2 Predictors from EPC data 36

6 Conclusions **37**

6.1 Future work 37

7 References **39**

A Data descriptions **41**

A.1 Land registry price paid attribute descriptions 41

A.2 EPC attribute descriptions 43

A.3 Index of Multiple Deprivation domain definitions 53

B Keras combined model summaries **55**

1 Introduction

The property industry in the UK is worth nearly £250 billion pounds annually [1] and the total value of residential property in the UK has been estimated at £7.39 trillion [2]. In 2018 there were over 1.2 million residential property transactions with a value of over £40,000 in the UK. Property and real estate is a very important part of the UK economy as higher house prices generally drive up consumer spending [3] as homeowners feel more confident in spending and are able to release equity against their homes. An increase in house prices also impacts the banking sector, as banks are more likely to lend mortgages at a time when house prices are rising, knowing they are more likely to have an asset that covers the debt in the event of a default. House prices are also a constituent part of the Retail Prices Index (RPI) in the UK, which can influence decisions on interest rates by the Monetary Policy Committee [4].

Property valuations are historically the preserve of estate agents with specialist knowledge of the market in a localised area and requires visiting the property to make an assessment of its value. Recently, online portals such as Rightmove and Zoopla have begun displaying estimated property values through automatic valuation models (AVMs) and many online services now exist offering semi-automatic valuations, in which the user must provide some details about the property before a value can be calculated. Other portals such as Homesearch and OnTheMarket are emerging that seek to augment or disrupt the traditional process of home buying by connecting buyers and sellers directly. If buyers and sellers are to be directly connected and the agent's role is reduced or removed, the valuations provided by online property portals may be the basis for the valuation of the house when it enters the market and as such they must be as accurate as possible. Having a good estimate of house prices is therefore valuable to individual buyers and sellers, organisations in the private sector and public sector bodies. Machine learning models may be an effective way to offer accurate estimates at large scale. This project arose from the identified need for improvements in AVMs and has two aims;

1. To determine which of the attributes that are publicly available in data from Energy Performance Certificate assessments are good predictors of property value

2. To determine whether visual features extracted from satellite images can be used as predictors to improve the performance of a neural network over

using tabular features alone

To achieve this, the project made use of multiple public data sources; Land Registry price paid data, Energy Performance Certificate assessment data, Index of Multiple Deprivation data and the Google Static Maps API. A number of processes were carried out on this data to obtain a usable dataset and machine learning techniques were used to create estimates that were then evaluated for comparison of the models.

2 Background

2.1 The Hedonic Pricing Method

The hedonic pricing method "relies on the assumption that a class of differentiated products can be broken down in to a number of characteristics. A combination of these characteristics and the external factors that affect the product determines the price" [5]. Property prices are a common example of this, where prices are determined by attributes of the property such as floor area, number of bedrooms and bathrooms, and garden size combined with characteristics of the surrounding area such as crime rates, access to amenities and schools. "HPM can be used to estimate the extent to which characteristics affect price by modelling property prices as a set of explanatory variables, including the structural, socio-economic and environmental characteristics... A limitation of HPM is that to obtain accurate and robust estimates large datasets with very detailed information about property characteristics and the surrounding environment are needed." [5]

2.2 Feed Forward Networks

A feed forward Artificial Neural Network (ANN) is a machine learning model that consists of layers of connected neurons. Layers include an input layer, one or more hidden layers and an output layer, and each layer consists of one or more neurons. Very often, each neuron in a layer is connected to each neuron in the subsequent layer, in what is known as a "fully connected" network, however this does not have to be the case and some of the architectures tested in this project are not fully connected, with some neurons and layers connected to a subset of the subsequent layers. The output of each neuron is determined by its inputs, a set of weights the same size as the set of the inputs, a bias and an activation function such that the output of each neuron is given by the formula $y = f(w_1x_1 + w_2x_2 + ... + w_nx_n + b)$ where y is the output of the neuron, f is the activation function, w_n is the weight applied to the input value x_n and b is the bias. The weights and biases are initialised randomly and then learned by the network during the model training process, while the activation function is chosen as part of the design and is often dependent on the application of the ANN. Activation functions such as the sigmoid function, tanh or the rectified linear unit (ReLU) are the main means by which neural networks are able to capture non-linearity in relationships.

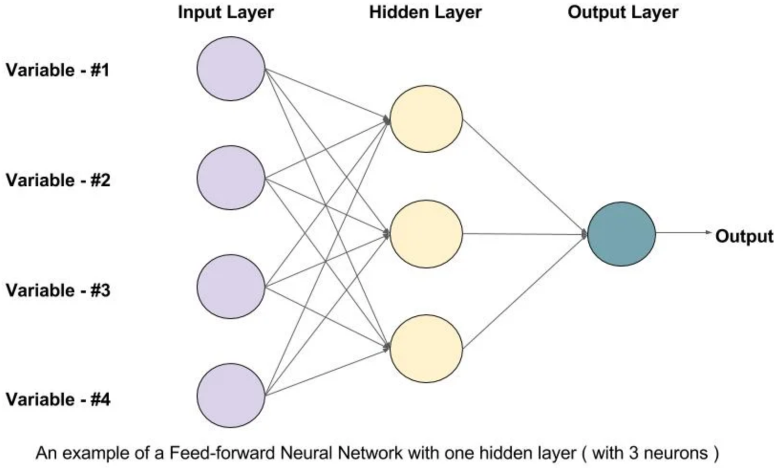


Figure 1: A feed forward network architecture [6]

The inputs to the first (input) layer are the features in the dataset and the inputs to subsequent layers are the outputs of the previous layer as shown in fig. 1. The output of the final layer is the prediction of the model. To learn the weights and biases, the model first makes a prediction or predictions using the initial, random set of weights and calculates the error or loss of that prediction from the known target values. The gradient (partial derivative) of the loss function is then calculated with respect to each weight and the direction in which the weight needs to be adjusted, with the goal of finding the minimum of the loss function - i.e. the point where the gradient of the loss function is zero - is determined. The weights are updated and the process is repeated. As the gradient becomes shallower, the magnitude of the weight adjustment is reduced to avoid overshooting the minimum. The process of iteratively adjusting the weights in the direction of the gradient towards the minimum is known as gradient descent. The weights are updated in this way from the final layer, backwards through the model to the first layer - a process known as backpropogation (backwards propogation of errors).

Several papers have explored the use of ANNs for estimating house prices including [7], [8] and [9], all of which achieved an improvement in measured error over the other methods used.

2.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of neural network that are very effective at image processing tasks such as object recognition or classification. CNNs employ convolution filters, pooling and non-linearity to reduce a matrix of data (such as the pixel values of an image) to a smaller dimensions which can then be understood by a feed forward network, which is usually implemented immediately after the convolution section (fig. 2) The convolution filters are small matrices (perhaps 3 x 3 or 5 x 5 for a filter with two dimensions) that are moved over the image and at each position, the dot product of the filter weights and the pixels in the image is taken to produce a single value for that patch of the image, thus convolving the image down to a set of extracted features. Fig. 3 shows an example of a possible filter configuration to detect vertical edges. Generally the dimension of the feature maps after convolution is given by $\frac{n+2p-f}{s} + 1$ where n is the size of the input image, p is the number of pixels of padding, f is the size of the filter and s is the stride (the number of pixels by which the filter moves at each step). Padding is added to allow the pixels at the edge of the image to be considered as important as the pixels in the middle. Without padding, the pixels at the edges of the image are not acted upon by the filter as many times as other pixels. This also has the effect of creating feature maps with the same dimensions as the input image.

Pooling layers aggregate the values in a matrix over a specified shape such as a 2 x 2 grid. The most commonly used aggregate function is the maximum, known as max-pooling. Non-linearity is applied in the same manner as in feed forward networks, using an activation function to transform the pixel values. When training a convolutional neural network, the network learns not only the weights in the feed forward section, but also the filter weights by the same process of gradient descent and backpropogation described in section 2.2.

[11] successfully used a convolutional neural network to extract features from

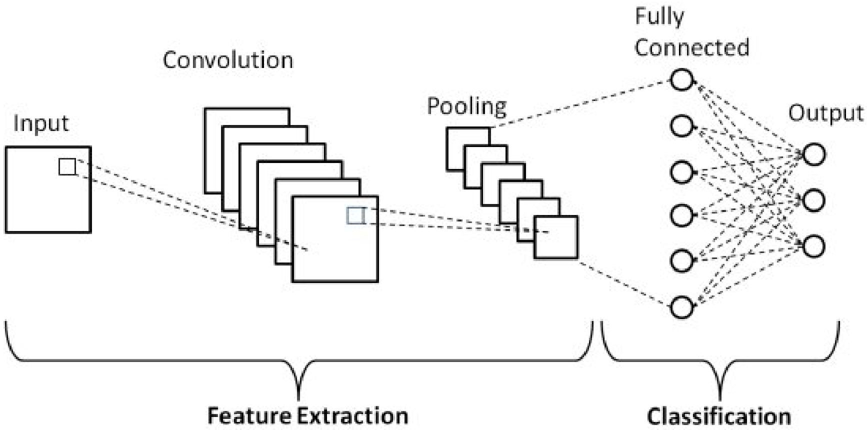
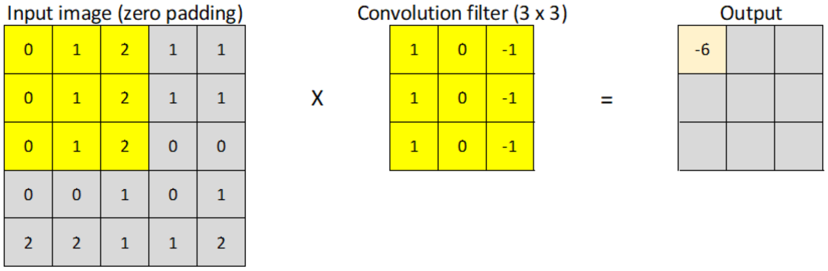


Figure 2: An example CNN structure for a classification task [10]



Input image (zero padding)

0	1	2	1	1
0	1	2	1	1
0	1	2	0	0
0	0	1	0	1
2	2	1	1	2

X

Convolution filter (3 x 3)

1	0	-1
1	0	-1
1	0	-1

=

Output

-6		

Figure 3: The 3x3 convolution filter is applied to the yellow 3x3 patch in the input image and the ouptut is shown on the right

Google Street View Images and Google Maps images in order to improve house price estimates.

2.3.1 Feature Extraction

A convolutional neural network performs feature extraction by default in any image processing task. As the model is trained, the filter weights are tuned to extract the features that best characterise or explain the target variable. The resulting feature maps are then flattened and fed into a feed forward network as described above. In feature extraction, the feed forward layers are stripped away, leaving the flattened feature maps as the final output of the model. These features are usually long, one-dimensional vectors that form a numerical representation of an image that can be understood by a fully connected network. The model can either be trained for a specific task, or a pre-trained set of weights can be used to extract features. The latter process is an example of transfer learning.

2.3.2 Transfer learning

Due to their complexity and number of trainable parameters, training a large and complex convolutional neural network requires large amounts of computing time and processing power. Such resources are often inaccessible to many and as such it is often not possible to develop and train a large and effective convolutional neural network from scratch for a specific task. Instead, it is possible to use a pre-trained model such as ResNet50, AlexNet or GoogLENet to recognise features in an image or set of images. Such networks are often available with weights pre-trained on the ImageNet dataset. ImageNet is a large dataset of over 1 million images that is widely considered the gold standard for assessing the skill of convolutional neural networks in image classification tasks. It is then possible, using a smaller amount of computing resources and time, to fine-tune the weights of such a model to a specific task. This may be done by making very small adjustments to all of the layers in the model, or only some of the later layers.

3 Methodology

3.1 Data gathering cleaning and preparation

3.1.1 Data gathering

The Land Registry price paid data was downloaded from the Land Registry web site. Full data was retrieved for the whole of England and Wales over the 24 month period from 1st Jan 2018 to 31st Dec 2019 inclusive. This data includes transactions for non-residential properties and “additional” price paid data, not related to the transfer of deeds to the property. This data was filtered to cover 5 postcode districts; HP10, HP11, HP12, HP13 and HP15, which cover the main settlement area of High Wycombe, Buckinghamshire, excluding surrounding villages. This dataset includes the price paid information that is the target variable for the machine learning models in this project. Other fields of interest include the general type of the property (detached, semi-detached, terraced or flat), the type of sale (freehold or leasehold), a new build flag (Y/N) indicating whether the property is newly built. The shape of this dataset was 3,675 rows x 16 columns.

Energy Performance Certificate data was downloaded for Wycombe district only, from the Ministry of Housing, Communities & Local Government. This file contained data relating to individual properties gathered during the energy assessment process for the issuance of Energy Performance Certificates in England and Wales. This includes but is not limited to the internal floor area of the property in square metres, information on the type of heating system, glazing, glazed proportion, floor height, number of extensions and energy efficiency ratings of various elements of the property such as the roof, walls and heating system. This dataset contained 57,146 rows and 91 columns.

It was also decided to download the Index of Multiple Deprivation data from the Ministry of Housing, Communities & Local Government. This data is aggregated over Lower Super Output Areas (LSOAs). LSOAs are small areas designed to be of a similar population size, with an average of approximately 1,500 residents or 650 households, to aid in the reporting of small area statistics in England and Wales. [12]. The Index of Multiple Deprivation measures several aspects of deprivation within each LSOA and assigns a score according to how deprived an area is in various sub-domains such as income, skills and education, crime, and barriers to housing and services. The version of the data used in this project also contains demographic data such as the size of the population over 60 years old. A higher score indicates greater deprivation. Each LSOA in the data has a score, a rank and a decile. A separate lookup table was also required to find a property's LSOA code from its postcode. The IMD data for the entire UK contains 32,844 rows and 57 columns.

Unique postcode values were extracted from the Land Registry data and passed as parameters in requests to the Google Maps Static API using the *requests* library in python. The API requests returned colour satellite images, centred over each postcode location, with zoom level 17 and resolution 600 x 600 pixels. One such image was collected for each postcode in the price paid data.

A full description of the attributes in each of the tabular data sources can be found in appendix A to this report.

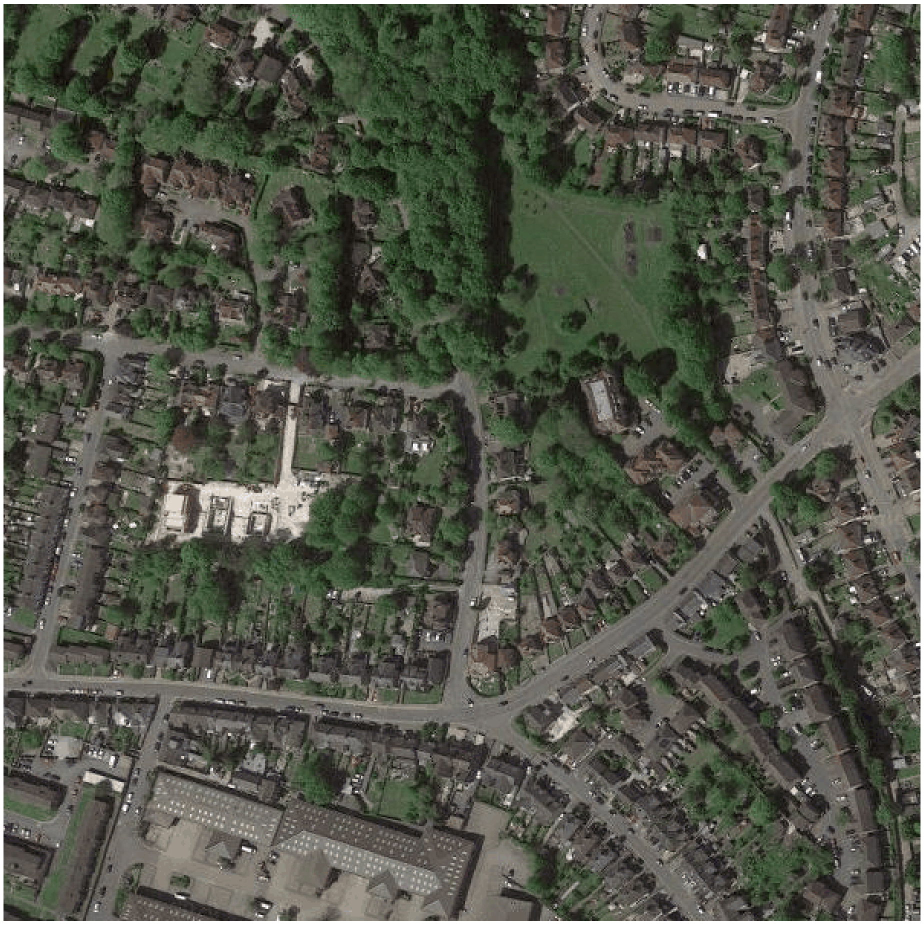


Figure 4: An example satellite image from the Google Maps API

3.1.2 Data merging

The first step in preparing the data was to merge the datasets together. For the EPC and Land Registry data this was achieved by matching addresses. The address data were arranged differently in each table so simple concatenation of the building number and postcode was used to create a unique address - as in theory no two properties will have the same number in the same postcode.

Initially, a regular expression was used to extract the most common patterns for building numbers from the first part of the address in each table, including the word “Flat” if present, so that “Flat 6” would be distinct from “6”. This number was then concatenated with the postcode and used for matching. This method returned 2,811 matches, but upon inspection several of these were incorrect where the building number followed the building name in the case of flats in the Land Registry data.

After inspecting the arrangement of the address fields it was determined that concatenating address fields in the Land Registry data in a certain order, combined with character replacement could be used to form a full address string which followed the same structure as, and could be matched to, the full address field in the EPC data. This method returned slightly fewer matches at 2,660 - however manual inspection of the matched data set revealed no mismatched records so this approach was chosen instead.

The Index of Multiple Deprivation data was joined through the postcode to LSOA lookup table, and only observations that had a match in the lookup were retained, resulting in 2,574 records.

The merged dataset was then split into 70% training and 30% test data to ensure that all future decisions on feature selection, cleaning and imputation were made on the training data alone and not influenced by the test data. The datasets were then written back to the MySQL database for use in future stages.

3.1.3 Data exploration

To understand the dataset better, some preliminary exploration was carried out to investigate the main relationships between the predictor variables and the target variable in particular. The main ways the data was expected to be distributed were visualised in simple charts.

As was expected, one of the strongest relationships was observed between total floor area and price (fig. 5). This is intuitive (larger houses cost more), however the strength of the relationship is notably high. This also revealed the outliers described in section 3.1.4. The distributions of both these variables was plotted (fig. 6). The distribution of both variables has a long right tail, confirming that most properties are close to the median value and very few properties sell for significantly more than this. The same is true of floor areas; most properties are of a similar size (approximately 90 - 100 sqm) and very few are much larger. Log transforming both these variables results in a more normal-looking curve, although the distributions were not tested for normality.

House prices move relatively slowly, perhaps low single figures of percentage change in a year. As the dataset covers 2 years of sales, there may be some error introduced by prices moving over time. The median price per square

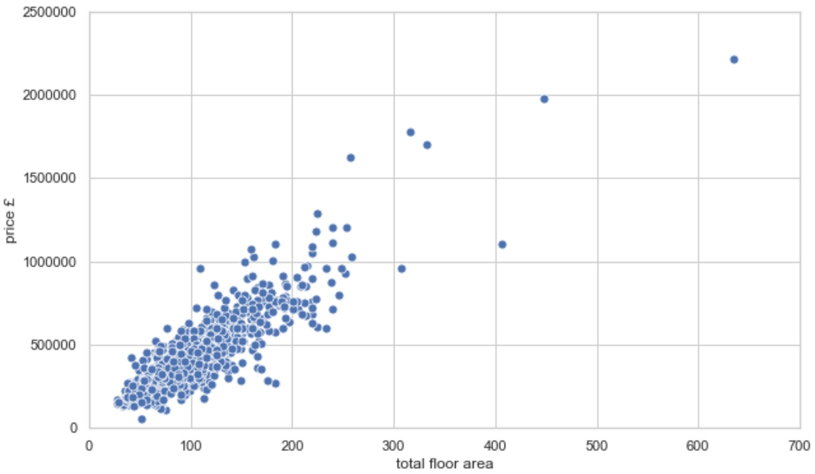


Figure 5: Price and floor area

metre by calendar quarter and year was inspected to check that there had not been too significant a rise or fall in prices over this period that could contribute to prediction error (fig. 7). There appears to be a slight decrease in average value adjusted for size over this period but it is not large. This also does not account for the different mix of properties sold in each quarter.

The Index of Multiple Deprivation provides data on how each LSOA in England and Wales scores in various aspects of deprivation. These aspects are averaged (with weighting) and combined to produce the Index of Multiple Deprivation score, which is then ranked from 1 to 32,844 with 1 being the most deprived area. High Wycombe is located in the relatively affluent area of South Buckinghamshire, however it is not as affluent as surrounding towns such as Beaconsfield and Gerrard's Cross and has some less affluent areas within it. This is reflected in the IMD data (fig. 8), which shows that the majority of the areas in High Wycombe have very high (less deprived) IMD ranks. However there are several areas with lower ranks; the minimum rank is 7,496 and the maximum is 32,834, suggesting that there should be some variation in prices that is influenced by attributes of the surrounding area, not just the property itself.

This is confirmed by the mean price in GBP per square metre by postcode sector. There is 33% variation between the cheapest and most expensive sectors

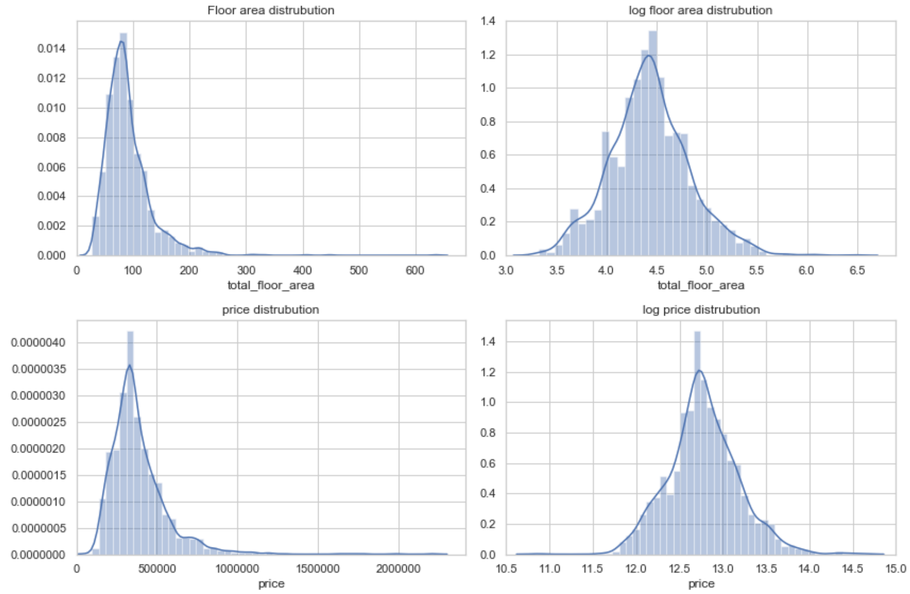


Figure 6: Distribution of price and total floor area; raw and log transformed

in the data (fig. 9)

3.1.4 Data cleaning and preparation

As the Land Registry data may contain repeat sales (i.e. properties that sold twice within the time period covered by the data), and the EPC data may contain multiple certificates for a single property, duplicates were removed to avoid weighting any individual property in the model more heavily. The record with the most recent EPC inspection date and most recent transaction date from the Land Registry data was kept for each unique address.

Administrative columns including but not limited to building reference number, row identifiers, addresses (excluding postcode) and transaction types were all removed as they were deemed not to be relevant to the value of the property. As the data relates to only one local authority, columns relating constituency and local authority were also removed.

After inspecting the values in the *built form* and *property type* columns, it was determined that *built form* was not interpretable as values such as “mid-terrace” and “end-of-terrace” were used to describe properties where the property type indicated a flat. Given that these columns likely duplicated the in-

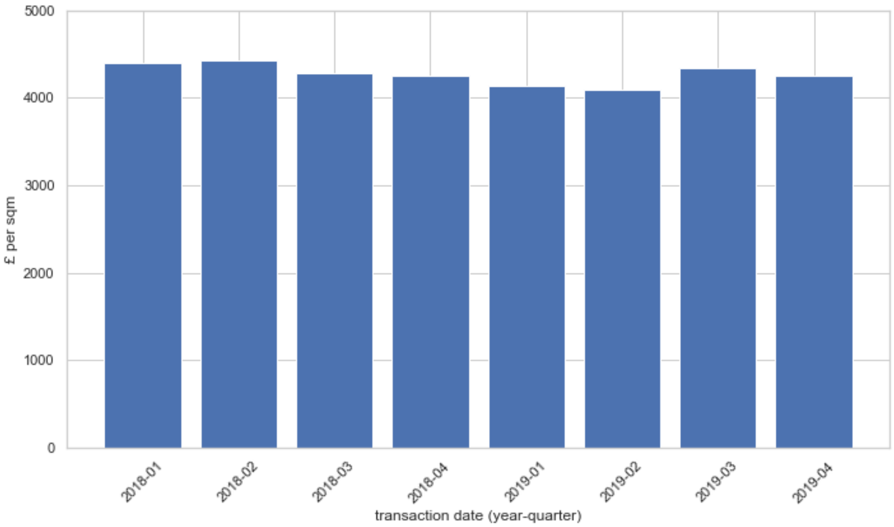


Figure 7: Median price per square metre by month and year

formation in the *ptype* (property type) column and their lower interpretability, these columns were also removed, however a one-hot encoded bungalow flag feature was created from this data prior to removal, indicating whether the property is a bungalow as this information is not contained in the Land Registry data.

The missing values in the dataset were visualised as shown in fig. 10. A large number of columns had a very high proportion of missing values. It was decided to remove all columns with more than 20% of values missing. The EPC data contained a fairly large proportion of columns with over 20% missing data, and a number of columns were discovered to be of poor quality or unusable.

Most significantly, several of the columns were description columns, which upon inspection contained in some cases over 90 unique values and varied in the information presented. For example, the *roof description* column sometimes stated the thickness of the insulation on the roof, sometimes just the type of insulation and sometimes the average thermal transmittance value in mW/h. Most of these attributes with this kind of variation could not be meaningfully grouped into a smaller number of categories, and in many cases there were a large number of missing values in these columns. Given the difficulties in utilising these features the decision was taken to remove all description columns from the data, with the exception of windows description.

Several records with floor areas of 0 square metres were identified and re-

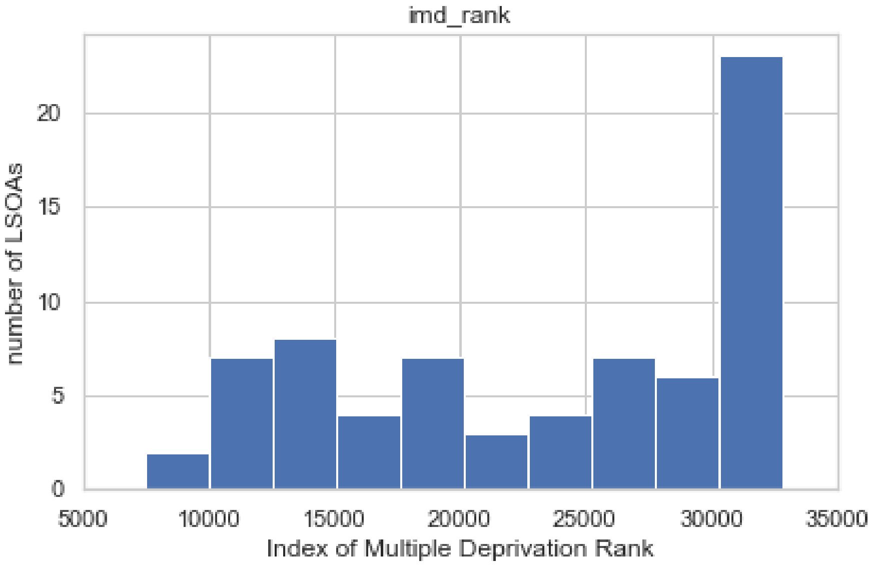


Figure 8: Distribution of Index of Multiple Deprivation ranks

moved from the dataset. One property with an abnormally high sale price was investigated, and further research indicated that this sale price had been entered incorrectly at the Land Registry as £4,920,000 while a very similar property next door sold a few months previously for £475,000. This price was therefore manually amended to £492,000.

Multiple aspects of the property such as the floors, roof, heating and windows were rated in the EPC data according to their environmental impact and energy efficiency on a scale with 5 levels from Very Poor to Very Good. Such ordinal variables were encoded with corresponding numbers from 0 to 4, then their correlations were plotted (fig. 11). Each aspect of the property had an environmental and energy efficiency rating and after calculating the Pearson correlation score for these variables it was determined that the two ratings for each aspect were always perfectly or extremely highly correlated. The environmental efficiency ratings were therefore dropped, leaving only the energy efficiency ratings.

The IMD data contained score, rank and decile for all the sub-domains on which the LSOAs are assessed. The decile is simply the rank grouped into 10 chunks so duplicates the information in rank, while rank is derived directly from

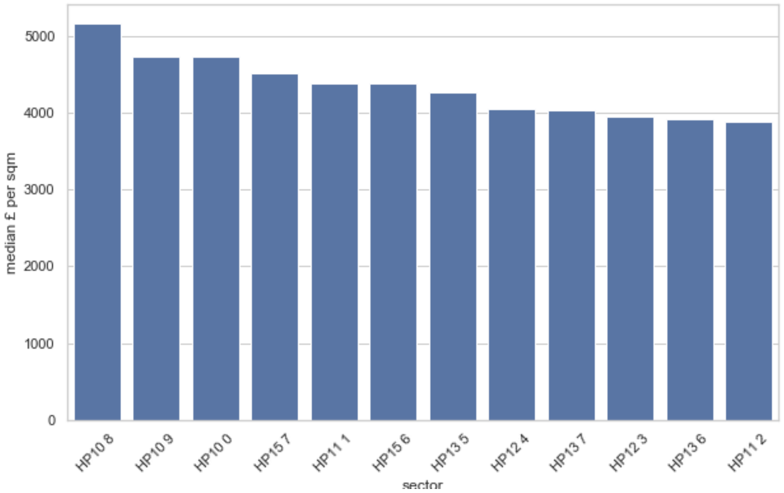


Figure 9: Mean GBP per square metre by postcode sector

the score. It was therefore unnecessary to retain all three versions of the same information in the data. It was elected to keep the score columns only, as it was decided that the score preserves the ratio of deprivation between areas better than rank - e.g. the difference in score between ranks 30,000 and 30,001 could be much greater than the difference in score between ranks 1,000 and 1,001.

Finally, the Pearson correlation score of all remaining numeric variables with the target variable was calculated, as well as the correlation matrix for the independent variables. Highly correlated variables with correlation score greater than 0.7 were removed from the data, retaining the variables with the highest correlation to the target.

After the cleaning process, the final dataset consisted of 2,567 samples (1,796 training examples and 771 test examples) and 21 predictor variables.

3.1.5 Data preprocessing

Simple imputation was carried out to fill missing values. As all the variables with missing values were ordinal or nominal, they were filled with their respective modal values.

As total floor area and the target variable were both found to be highly right-skewed and highly correlated, both were log-transformed. More evenly

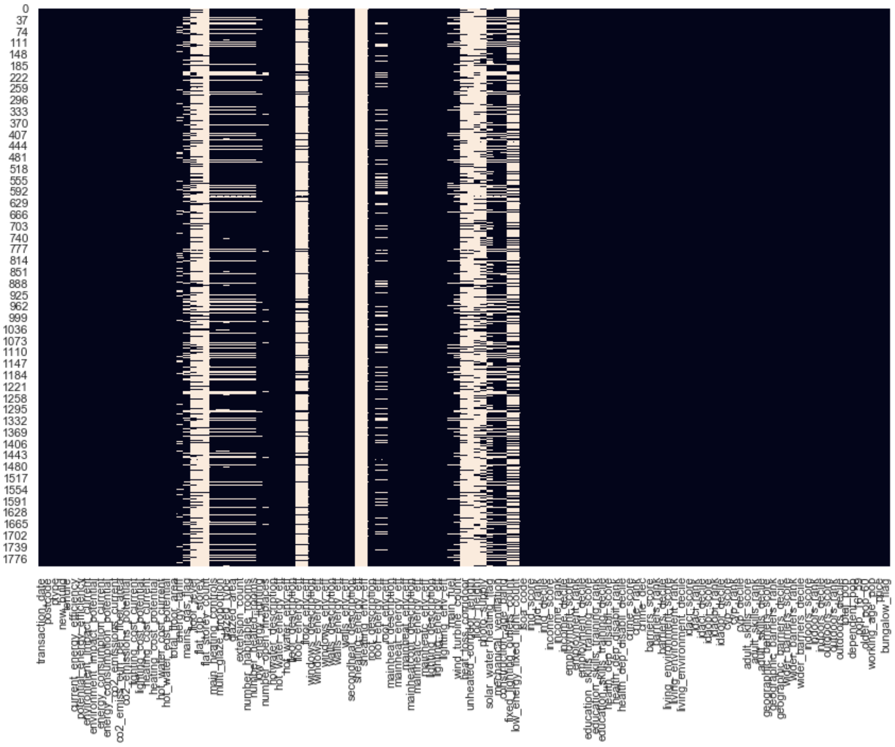


Figure 10: There were many missing values in the EPC data

distributing the values of variables in this way is beneficial for many machine learning algorithms including neural networks as it avoids the weight of the variable being reduced due to the data points being compressed to one end of the scale.

The remaining nominal variables (construction age band and windows description) were one-hot encoded and numeric variables were scaled between 0 and 1.

The result was a training dataset with 1,796 rows and 30 columns including dummy variables.

3.2 Feature selection

Cross-validated recursive feature elimination (CVRFE) was applied to the training dataset using an ordinary least squares (OLS) model using R-squared as the evaluation metric. In this process a model is built using all the variables and

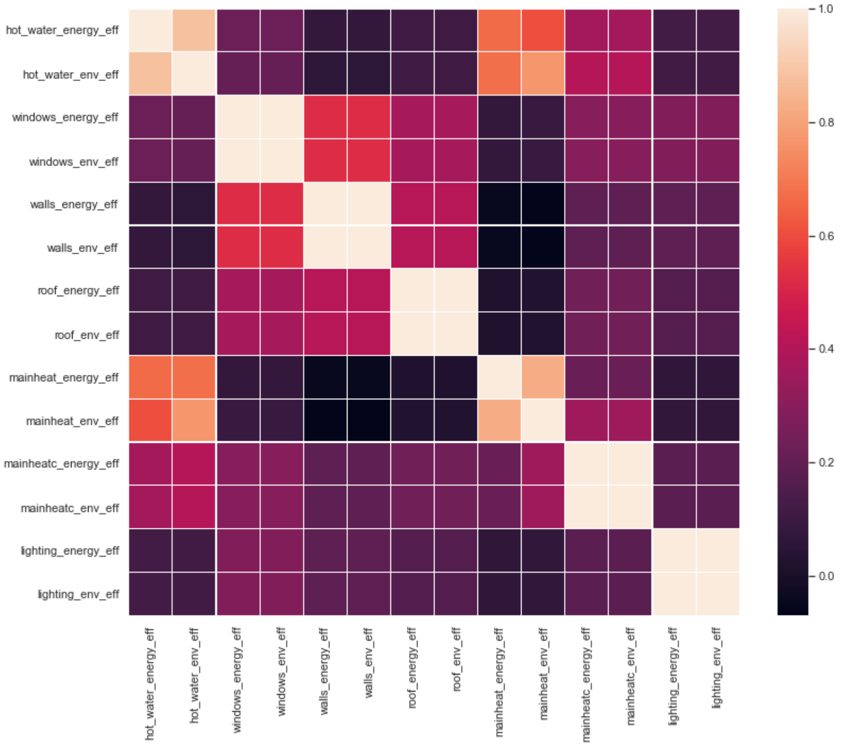


Figure 11: Correlation between energy and environmental efficiency columns

the weakest predictor is removed from the data. This process is then repeated with the remaining variables until a specified minimum number of variables is reached (in this case 4). The models are cross validated on the training data at each step and the cross validation error is used to determine the optimum number of features to retain. This resulted in a 25 out of 33 predictor columns being selected.

P-values cannot be obtained from the class used for this process so could not be reviewed, but the coefficients of the retained variables were found to be very small, meaning little influence on price, and counter-intuitive in some cases. It was decided to also create a manually curated dataset with only the features with the strongest influence on price (coefficients above 0.05), as it seems likely the model had found some spurious correlations as a result of a quirk of this particular dataset. Removal of variables with coefficients below 0.05 removed the majority of the counter-intuitive coefficients, and manual removal of remaining

Table 1: Feature coefficients

Feature	**Coefficient**
total_floor_area	1.678863
ptype_F	-0.397136
ptype_T	-0.234404
cyp_score	-0.199376
ptype_S	-0.170198
new_build_Y	0.168566
energy_consumption_current	-0.138235
mainheatc_energy_eff	0.123792
older_pop_60	0.122135
number_open_fireplaces	0.112167
barriers_score	-0.090625
construction_age_band_England and Wales: befor...	0.063074
mains_gas_flag_Y	-0.041172

variables that were judged to be unlikely to be good predictors left 13 features which are shown in table 1 with coefficients determined by the OLS model re-fitted on the whole training dataset.

One of the aims of this project was to determine which, if any, of the attributes in the EPC data could be used as predictors of price and the results in table 1 are discussed in more detail in section 5.2.

3.3 Baseline models

3.3.1 Linear regression model

Two ordinary least squares models were trained on the training data using both the full set of 25 CVRFE selected features and the reduced subset of 13 manually curated features. Both were then used to predict values on the test data.

3.3.2 Random Forest

A random forest model was constructed with basic hyperparameters again selected using a grid search cross validation approach. The parameters searched were maximum depth, number of trees and minimum samples per leaf. 45 combinations of parameters were tested in total. The best combination of parameters was found for each of the two datasets, then these parameteres were used to fit the model on the whole training set each time.

3.3.3 Baseline neural network model

To build the neural network, it was chosen to use the ReLU activation function for both the hidden and output layers as it has been shown to perform well on regression problems, while other activation functions such as sigmoid or tanh are usually unsuitable due to their compressing effect. Usually a linear activation function would be used on the output layer for a regression problem, but in this case the estimated value should never be negative so ReLU appeared the better option. It was also chosen to use the Adam optimizer as it performs well on a wide variety of problems. Dropout was applied after the input layer to prevent overfitting, although during hyperparameter tuning the best value for this parameter was consistently zero and no dropout was actually applied in the final model.

As the number of predictors is not large and the complexity of the initial model is relatively low, it was decided to employ a single hidden layer to capture any non-linearity in the relationship between predictor and target variables.

A grid search cross validation approach was used to select optimum hyperparameters for batch size and number of neurons in the hidden layer. The best parameters were selected on the mean absolute error after training the model for 100 epochs. This process was carried out on both the 25-feature and 13-feature datasets. The best hyperparameters found are shown in table 2 with the associated cross validation errors.

Dataset features	**Batch size**	**Neurons in hidden layer**	**Cross validation error**
25	20	128	0.118
13	10	64	0.112

Table 2: Best parameters and cross validated training error

The optimum number of epochs was then selected by training the models again for 300 epochs each and plotting the validation loss history to find the point at which no further improvement in mean absolute error was made (fig. 12).

After training and testing the neural networks for the optimum number of epochs, the test error achieved was slightly higher than both the OLS and

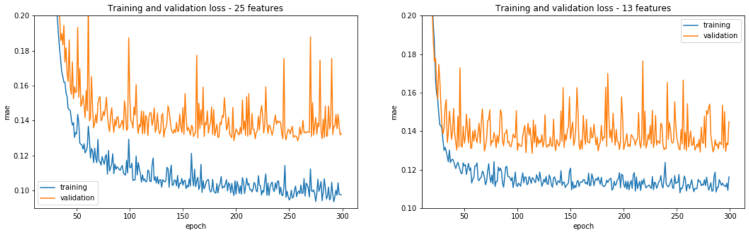


Figure 12: Training and validation loss on the 25-feature and 13-feature datasets

random forest models on both data sets. This may be partly due to the high variance associated with the random initialisation of weights in a neural network model. For this reason an ensemble effect was simulated by training the models with a different random initialisation for 10 iterations and predicting the test set each time, taking the arithmetic mean of the predictions to be the final predicted value. This approach yielded a slight decrease in mean absolute error from 0.12 to 0.11 for the models trained on both data sets. This brought them roughly in line with the linear regression model and only slightly worse than the random forests. R-squared and adjusted R-squared values were also calculated for each model. Adjusted R-squared is particularly interesting when comparing between datasets with different numbers of predictors, as is the case here, as it punishes the inclusion of extraneous variables in the model.

3.4 Visual features

3.4.1 Data preparation

To extract features related to house prices from satellite images, a dataset was created that contained one record per postcode comprising the image data as a numpy array, and the corresponding target variable being the mean of the sale price of properties in the corresponding LSOA for each postcode in GBP per square metre. It was decided to use price per square metre as the response variable in order to control for the size of properties in the area, which was found while building the first model to be by far the most influential variable in a property's value. This price averaging was carried out to reduce the noise in the data; as properties transact infrequently, the number of properties sold in any given postcode in the two year period covered by the data is very small; perhaps one or two properties per postcode. Given that the value of the property is also informed by its own attributes, this can result in apparent large variances in price per square metre between neighbouring postcodes. Taking the average of a widened area gives a more stable indication of whether the postcode is in a cheaper or more expensive area.

The image data was converted to numpy array format using the *Image* class from the *pillow* library. The original PNG files contained 4 channels - red, green, blue and alpha - of which only the RGB channels were used.

Rather than using a completely random method to split the data into training and test sets, it was necessary to separate the data in a geographical manner to ensure that any success of the model was not due to its ability to simply memorise the images and recall the correct output [1]. The data was therefore split by randomly selecting 15% of LSOAs to become the test set.

In addition to reducing noise, this process of averaging and splitting by LSOA was intended to ensure that:

1. There is no contamination of the training data by including sales from the test data in the averaged values

2. The data is geographically separated

It was decided to use only 15% of LSOAs as test data as the number of unique postcodes (and therefore images) in the sales data was discovered to be much lower than anticipated. Out of the 2,511 observations in the sales data there were 1,192 unique postcodes in total. This resulted in 1,024 training examples and 170 test examples.

2 observations in the data were removed as outliers, having an average per square metre cost over 50% greater than the next nearest observation. In each case, there was only one postcode in the LSOA contained in the dataset.

The Google Maps Static API did not return an image for postcode HP15 7TH, but rather a blank square with an error message. So this was replaced with the image for HP15 7TP, the physically closest postcode in the dataset.

This data was stored as a pickle file to be loaded into Google Colab where GPU acceleration could be utilised to speed up the training process of the CNN.

The image size was reduced from 600 x 600 to 128 x 128 pixels, as there were memory issues when using the full sized images. A bug in the Keras API was also encountered that resulted in doubling the RAM usage when training the model, although this was later resolved after much work had already been completed with the images in 128 x 128 resolution, so this resolution was maintained given time constraints.

3.4.2 Model selection

3 existing neural network architectures were considered as feature extractors; ResNet50 [13], InceptionResNetV2 [14] and DenseNet201 [15]. ResNet50 was chosen as it has been shown to be effective in the classification of satellite imagery [16] and could therefore also perform well on a regression problem with satellite imagery. InceptionResNetV2 is a variant of ResNet50 that achieved superior performance on the ImageNet dataset over ResNet50, while DenseNet was included as a wildcard because of its very different architecture to the other two models, and its smaller number of parameters than ResNet. [15]

The top layers of each model were removed and a 2d global averaging layer was added to convert the convolution blocks into a single vector of features - those that would later be extracted by the chosen model. A dropout rate of 50% was then applied before single output neuron with relu activation to output the prediction value.

A simple CNN was also constructed to be trained from scratch, based on LeNet architecture [17], consisting of 3 each of convolution with filter sizes 3, 5 and 7 respectively, 2x2 max pooling with strides of 2, and 10% dropout after each max pooling layer. The relu activation function was used for each layer. The final 3 layers were global averaging, 50% dropout and a single output neuron - in that order - as with the pre-trained models. This architecture was designed with the goal of minimising the number of trainable variables in order to avoid overfitting to the training data, given its small size, after early experiments with the CNN suggested this might be the case. Zero padding was also used in an attempt to reduce the importance of the data at the edges of the image, where images are more likely to overlap.

A validation set was also created, geographically separated in the same manner as the train/test split, by selecting a further 15% of LSOAs in the training data to be used as validation LSOAs. The training and validation sets consisted of 905 and 170 test samples respectively.

As the size of the final training dataset is very small, data augmentation was applied during training with random rotation between 0 and 90 degrees, horizontal and vertical flipping to reduce the level of overfitting. Each model was trained for 30 epochs and callback was used to write the model state to

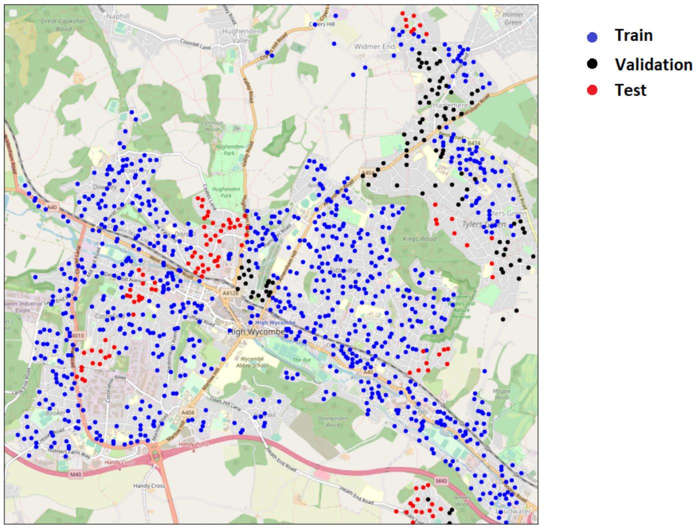


Figure 13: Train, validation and test locations

disk every time the validation loss reached a new minimum, thereby retaining the best number of epochs.

The models were assessed on their skill at predicting the average price per square metre assigned to the postcode as described above, on the basis that this gives an indication of how well the feature vectors can be related to price. Models were assessed on both mean squared error and the Pearson correlation coefficient between the predicted and the true values, as the most important thing at this point was that more expensive areas be generally predicted to be more expensive and vice versa, rather than any particularly accurate estimate of the output variable, as the target variable itself is an approximation.

The models' scores are shown in table 3. Only two models produced positive correlation with the target variable, the basic CNN and ResNet50. These models were chosen as the feature extractors to be tested in the final model.

Model	**mse**	**correlation**	**lsoa_avg_correlation**
BasicCNN	444.85	0.23	0.29
ResNet50	895.29	0.14	0.00
DenseNet201	1053.94	-0.10	-0.30
InceptionResNetV2	1664.49	0.07	-0.06

Table 3: CNN model scores

3.4.3 Feature extraction

To extract features, the best saved weights of the basic CNN were loaded and the final layer (the prediction neuron) was removed, leaving the global averaging layer which produces a single feature vector. Predictions were then made on both the training and test dataset and the resulting feature vectors were stored as pickle files for use in the final stage of the model.

The ResNet50 model was not fine tuned due to time and computing power restraints in this instance, so the features from this model were extracted using the weights pre-trained on ImageNet. Again a global averaging layer was added to produce a single feature vector from the 5 x 5 x 2048 output of the ResNetModel.

ResNet50 produces feature vectors of length 2048 while the basic CNN vectors contain just 65 elements, indicating that the neural network architectures for each feature set would need to be significantly different in the final models.

3.5 Combined feature models

To produce the final set of models for comparison, the tabular data was re-split by LSOA to match the split used in the feature extraction stage, to avoid leakage between the training and test features.

Only the 13 feature dataset was used as the results in section 3.3 showed little to no difference in mean absolute error using this data against the 25 features selected by cross validated recursive feature elimination. Using the smaller dataset reduces model complexity and reduces the likelihood of overfitting.

The baseline models were re-trained and tested on the new data split to give the baseline scores. This step could have been avoided if the need for the geographic split had been anticipated and the data had been split that way from the start, however the models should perform similarly well on any subset of the data given that their parameters were chosen using a cross-validated approach, so it is not considered to be a significant issue.

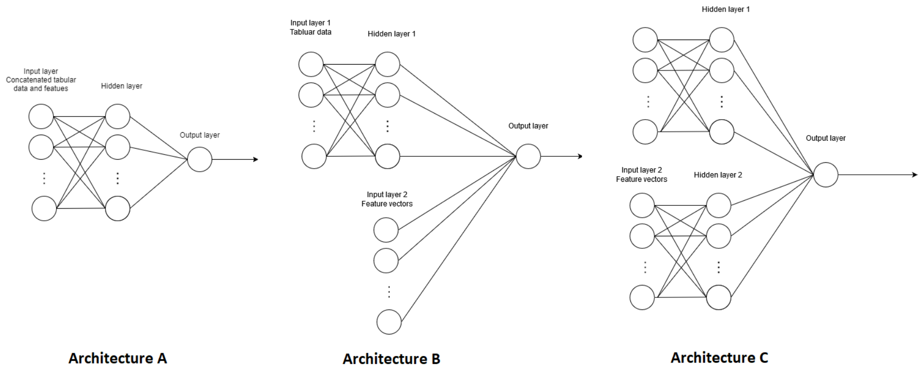


Figure 14: Model architectures

Several ANN architectures were evaluated for each of the basic and ResNet50 feature sets, both using a single input taking the concatenated tabular and visual features as one table, and using two separate inputs to pass the tabular and visual features through different layers of neurons for greater flexibility. The general configuration of each architecture is shown in fig. 14. The Keras model summaries of each architecture are contained in appendix B to this report.

The ResNet50 features were found to contain several columns that were all zero values which were removed prior to training, resulting in 1,390 features from the original 2,048 being used.

4 Software and packages

The below outlines the major software and libraries used in this project and their applications. Where a different version was in use on Google Colaboratory, this is shown as the second version number e.g. 1.1.0/1.1.5 indicates that the local version was 1.1.0 and Google Colaboratory uses version 1.1.5 of the library.

- **Python 3.7.3** - The base language in which the vast majority of work was undertaken. It was chosen for its user-friendly syntax and the wealth of machine learning libraries that are available for the language.
- **MySQL 8.0** - Tabular data was stored to a MySQL database to allow easier storage of multiple tables and allow joining between them.

- **Keras 2.3.1/2.4.3** - The main library chosen for creating neural networks using both the Sequential class, which was used to implement the baseline neural network model, and the Keras Functional API which was used to create the final models that are not fully connected. Keras also provides the pre-trained model weights and architectures for the CNNs tested in the model selection phase (section 3.4.2).
- **tensorflow 2.1.0/2.3.0** - The backend for Keras.
- **scikit-learn 0.22.1/0.22.2.post1** - Mainly utilised for cross validation and parameter search tasks. Provides a wrapper for Keras models to make them compatible with cross validation functions and classes in scikit-learn.
- **Pandas 1.0.5/1.1.2** - The Pandas API was used for extracting data from the database and for the majority of data manipulation and aggregation tasks in Python.
- **numpy 1.18.1/1.18.5** - Storing image data as arrays and some data manipulation.
- **seaborn 0.10.0/0.11.0** - data visualization
- **matplotlib 3.1.3/3.2.2** - data visualization
- **Google Colaboratory** - Google Colaboratory (Colab) is a cloud-based Python environment with a notebook interface. All work relating to convolutional neural networks was undertaken on Colab as it offers free (but limited) access to a GPU, providing vastly superior performance for training and testing CNNs over using a local CPU.

A complete list of installed libraries can be found in appendix C to this report

5 Evaluation

The primary aim of this project was to determine whether features extracted from satellite imagery could be used to improve the accuracy of house price estimates in a neural network model. The secondary aim was to evaluate the usefulness of property features gathered during Energy Performance Certificate inspections in predicting property prices.

5.1 Model Accuracy

The main metrics used for evaluating the models were mean absolute error and R-squared value. Given that the target variable was the natural logarithm of the sale price rather than raw sale prices, it is first necessary to understand the interpretation of the mean absolute error. Mean absolute error is given by the formula

$$\frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

The mean absolute percentage error is given by the formula

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i}$$

In the case where we are predicting the natural logarith of the price, to calculate mean absolute percentage error we would first exponentiate the predicted and true values so that the mean absolute percentage error is:

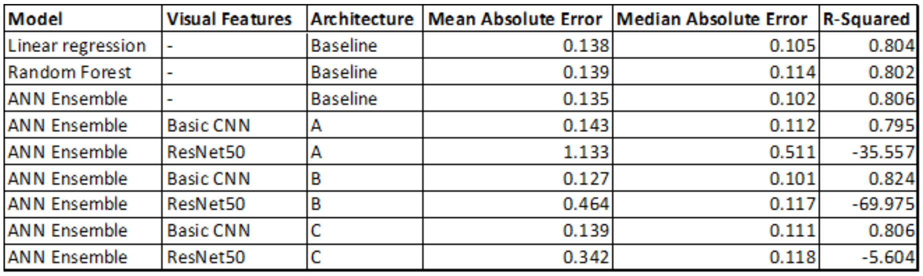
$$\frac{1}{n}\sum_{i=1}^{n}\frac{|e^{\hat{y}_i} - e^{y_i}|}{e^{y_i}}$$

Given that

$$\begin{aligned}&\frac{|e^{\hat{y}} - e^{y_i}|}{e^{y_i}}\\&= e^{|\hat{y}-y|} - e^{|y-y|}\\&= e^{|\hat{y}-y|} - 1\end{aligned}$$

it can be seen that $\hat{y} - y \approx e^{|\hat{y}-y|} - 1$ for small values of $\hat{y} - y$. While at first glance this seems needlessly complex, the simple result is that the MAE of the log predicted price can be interpreted as loosely approximating the mean absolute percentage error, however the MAE of log price is non-linear and is less influenced by larger errors. This was seen as an advantage because of the small size of the test set and, as there is no existing research on this dataset to which meaningful comparisons could be made, it was the simplest approach to evaluating the model to use the MAE from the log predictions. This was also preferable to calculating the MAE in raw GBP as this doesn't account for the value of the properties. An error of £40,000 may not be a good estimate for a property worth £100,000 but would be very good for a property with a value of £1,000,000.

The baseline models performed slightly worse on the new test/train split of the dataset, with the ANN ensemble method achieving the lowest mean absolute error of the three at 0.132.



Model	Visual Features	Architecture	Mean Absolute Error	Median Absolute Error	R-Squared
Linear regression	-	Baseline	0.138	0.105	0.804
Random Forest	-	Baseline	0.139	0.114	0.802
ANN Ensemble	-	Baseline	0.135	0.102	0.806
ANN Ensemble	Basic CNN	A	0.143	0.112	0.795
ANN Ensemble	ResNet50	A	1.133	0.511	-35.557
ANN Ensemble	Basic CNN	B	0.127	0.101	0.824
ANN Ensemble	ResNet50	B	0.464	0.117	-69.975
ANN Ensemble	Basic CNN	C	0.139	0.111	0.806
ANN Ensemble	ResNet50	C	0.342	0.118	-5.604

Figure 15: Model results

The models trained on the combined tabular and visual features dataset generally had higher mean absolute error than the baseline models, especially the models trained with the ResNet50 features which achieved negative R-squared scores and very high mean absolute error, although it is not clear why they performed so poorly as this.

The models using the basic CNN extracted features performed better than their ResNet50 counterparts, and architecture B even showed a slight, although not very significant improvement in mean absolute error. This was surprising as the expectation was that the more advanced model would have superior performance, however there are two things to consider:

1. The ResNet50 model suffered from not being fine tuned for this task. This was due to the learning process of the researcher undertaking the project and available computing resources. The basic CNN benefited from having its convolution layers trained on this specific dataset for this specific task. Had the fine tuning been carried out on ResNet50, its performance may have been improved.

2. It is possible, given the close physical proximity of some of the postcodes in the training and test datasets, that the CNN was able to recognise significant and features in nearby areas, more akin to recalling them from memory rather than generalising about features that influence prices.

With regards to the second point, it may be that the settings used in retrieving the photographs were detrimental to the experiment. When making

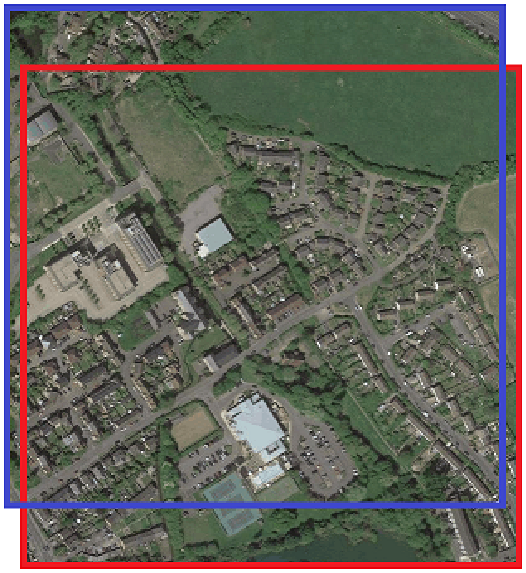


Figure 16: Overlapping area of nearest images from train (blue) and test (red) data

calls to the Google Maps Static API, a zoom level of 17 was specified, as it was felt that this gave a sufficient view of the area immediately surrounding the postcode location without encompassing too wide an area. However, it may have been better to use a tighter field of view around the postcode, to enhance the resolution of localised features and reduce the amount of overlap between photos, reducing the model's ability to memorize features and force it to find a a set of filters and weights that extract features that better generalize. To investigate the extent of the overlap, latitude and longitude information for postcodes in the HP postal area were obtained and the distances between the test and training postcodes were calculated using the haversine formula. The images for the postcodes with the shortest distance between them (HP10 0FH and HP10 0FL) were overlayed to identify how much of the image was shared (fig. 16). By measuring the pixels in the overlapping area it is estimated that in this worst case, approxiamtely 85% of the image area is shared between the train and test images. Despite efforts to keep the train and test data separate it was not possible to keep them completely separate while still obtaining a reasonable cross-section of the town.

The overlap is obviously significant, but the example shown here does repre-

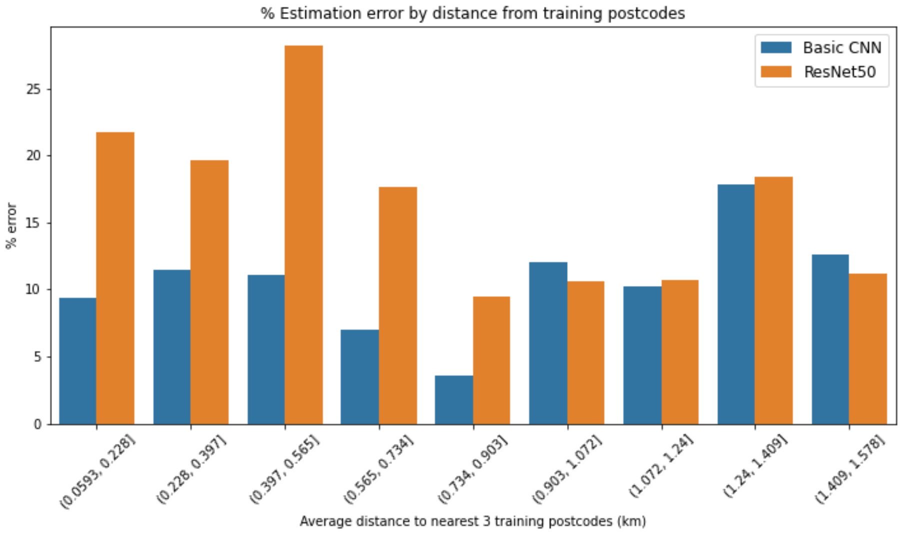


Figure 17: Average prediction error by distance

sent the worst case present in the data. To analyze the possible effects of such image overlap, the relationship between the physical distance from the training data postcodes and the prediction error during the model selection phase for each test postcode was examined. If the error was consistently lower closer to the training data, it suggests that the filters may be trained to find overlapping areas of the images and predict similarly for them, while failing to predict accurately for images further from the training set where little to no overlap occurs. Figure 17 shows the result of this analysis and it did not appear to be the case that error increased with distance from the training data.

The small number of observations in the dataset was also a slight concern throughout, although it is larger than the dataset used by Ahmed and Moustafa [18]. High Wycombe was chosen for this project as the researcher is familiar with the housing market there which it was aniticipated would be useful, however this knowledge did not in the end provide any great benefit and it may have been wiser to chose a larger town to increase the amount of available data. This highlights another potential pitfall with neural networks for house price estimation; they work best with large amounts of data. To obtain enough data it is necessary to obtain sales information spanning several months or even years which is likely to mean the estimates are not a true reflection of up-to-date values. This could perhaps be overcome if data from across England and

Wales or the UK could be used, however this presents its own, new challenges that are beyond the scope of discussion in this project.

5.2 Predictors from EPC data

Immediately upon exploring the data, it became apparent that many of the variables that were hoped to be good predictors of price, such as the type of heating system, were of poor quality with many missing values and many non-interpretable values. Many features had to be discarded on this basis. There was also a high degree of collinearity between many of the numerical variables, largely because attributes such as energy consumption and heating costs are heavily determined by the size of the property.

The number of features that were actually taken through to the feature selection stage was just 20 out of 132 original variables which was disappointingly low. Some features however do appear to be useful predictors; the strongest by far being the the floor area of the property which has over 4 times the influence on price of the next strongest predictor.

One of the features that was kept in the manually curated dataset was *mainheatc_energy_eff* which is the rating of the energy efficiency of the main heat controls of the property. This seems intuitively unimportant, even more so than the energy efficiency of the heating system itself, however as the coefficient was relatively high, it was kept as a possible proxy variable. It is possible that the efficiency of the heat controls is an indicator of how modernised or how high-spec the property is and the level of sophistication of the installed systems, which could be an indicator of overall quality, thus influencing the price.

Conversely, the dummy variable for older properties in the construction age band ”pre-1900” also had a positive coefficient (0.063) which contradicts the idea that more modern properties are more expensive. However, it could be that older properties are perceived as having more character, and it is the character that is worth the extra premium. This is perhaps backed up by the fact that the number of open fireplaces has a relatively strong influence on price (0.112); while fireplaces themselves may not intrinsically hold much value, this could again be a proxy variable, this time for the character or grandness of the property.

6 Conclusions

It was possible to reduce the mean absolute error and R-squared score of a neural network model for predicting house prices in High Wycombe by extracting visual features from satellite images. There was a 6% reduction in MAE and a 2% increase in R-squared score over the baseline ANN model. The ANN model also outperformed the linear regression and random forest models built without the visual features. It is not entirely clear whether this is due to the neural network memorizing features that are shared between images or whether the features are well generalised, but it is possible to improve estimates over at least a small area using this method.

Very few of the attributes in the EPC data were useful predictors of value. Total floor area is by far the strongest predictor, while other property features added little information.

6.1 Future work

Possible future work identified during the completion of this project was mainly the result of limitations discovered at various stages. There are several areas where improvements could be implemented:

- The image resolution was reduced from 600 x 600 to 128 x 128 which could have produced artifacts in the image that would be picked up by the CNN as features. Using a larger image size and not resizing the images could improve performance.

- Using a higher level of zoom from the Google Maps API to reduce the overlap of the images would allow for better understanding of the performance of the model.

- Developing a method for separating the images into train and test datasets with no overlap would aid in assessing how well the model generalizes. Using two different towns as training and test data separately may be an option, however this brings its own set of unknowns and problem, such as whether the same features have the same influence on prices in both towns or if other factors have more influence in one area than another.

- Fine tuning the transfer learning models may yield better results, as may using different CNN architectures not considered in this project.

- An interesting extension of this project might be to combine it with the work of Ahmed and Moustafa [18] who extracted features from images of the property itself, and attempt to estimate property prices solely from image data, with no tabular information at all.

7 References

[1] Office for National Statistics, “House price statistics for small areas in England and Wales: year ending June 2019,” 2020.

[2] L. Bowles and S. Laming, “UK housing stock now worth a record £7.39 trillion after decade of gaining £750 million a day.” [Online]. Available: https://www.savills.co.uk/insight-and-opinion/savills-news/294601/uk-housing-stock-now-worth-a-record-£7.39-trillion-after-decade-of-gaining-£750-million-a-day. [Accessed: 02-May-2020].

[3] “How the housing market affects the economy - Economics Help.” [Online]. Available: https://www.economicshelp.org/blog/21636/housing/how-the-housing-market-affects-the-economy/. [Accessed: 15-Aug-2020].

[4] S. Lim and M. Pavlou, “An improved national house price index using Land Registry data,” *RICS Res. Pap. Ser.*, vol. 7, no. 11, pp. 1–29, 2007.

[5] Office for National Statistics, “Value of nature implicit in property prices – Hedonic Pricing Method (HPM) methodology note”, 2018 [Online]. Available: https://www.ons.gov.uk/economy/environmentalaccounts/methodologies/valueofnatureimplicitinpropertypric
[Accessed: 11-Apr-2020]

[6] “Understanding Feedforward Neural Networks — Learn OpenCV.” [Online]. Available: https://www.learnopencv.com/understanding-feedforward-neural-networks/. [Accessed: 20-08-2020]

[7] A. Bin Khamis, N. Khalidah, and K. Binti, “Comparative Study On Estimate House Price Using Statistical And Neural Network Model,” *Int. J. Sci. Technol. Res.*, vol. 3, no. 12, pp. 126–131, 2014.

[8] V. Limsombunchai, C. Gan, and L. Minsoo, “House Price Prediction: Hedonic Price Model vs. Artificial Neural Network,” *Am. J. Appl. Sci.*, vol. 1, no. 3, pp. 193–201, 2004.

[9] J. M. Núñez Tabales, J. M. Caridad Y Ocerin, and F. J. Rey Carmona, “Artificial neural networks for predicting real estate prices,” *Rev. Metod. Cuantitativos para la Econ. y la Empres.*, vol. 15, no. 1, pp. 29–44, 2013.

[10] V. H. Phung and E. J. Rhee, “A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” *Appl. Sci.*, vol. 9, no. 21, Nov. 2019, doi: 10.3390/app9214500.

[11] S. Law, B. Paige, and C. Russell,“Take a look around: Using street view and satellite images to estimate house prices,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 5, pp. 1–19, 2019, doi: 10.1145/3342240.

[12] “Census geography - Office for National Statistics.” [Online]. Available: https://www.ons.gov.uk/methodology/geography/ukgeographies/censusgeography. [Accessed: 25-Sep-2020].

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *31st AAAI Conference on Artificial Intelligence*, AAAI 2017, 2017, pp. 4278–4284.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *30th IEEE Conf. Comput. Vis. Pattern Recognition*, CVPR 2017, 2017-January, pp. 2261–2269, Aug. 2016.

[16] D. Gardner and D. Nichols, “Multi-label Classification of Satellite Images with Deep Learning”, *Unknown publication*, 2017.

[17] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, ”Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791

[18] E. H. Ahmed and M. Moustafa, “House price estimation from visual and textual features,” IJCCI 2016 - Proc. 8th Int. Jt. Conf. Comput. Intell., vol. 3, November, pp. 62–68, 2016, doi: 10.5220/0006040700620068.

A Data descriptions

A.1 Land registry price paid attribute descriptions

Attribute	Description (where appropriate)
Transaction unique identifier	A reference number which is generated automatically recording each published sale. The number is unique and will change each time a sale is recorded.
Price	Sale price stated on the transfer deed.
Date of Transfer	Date when the sale was completed, as stated on the transfer deed.
Postcode	This is the postcode used at the time of the original transaction. Note that postcodes can be reallocated and these changes are not reflected in the Price Paid Dataset.
Property Type	D = Detached, S = Semi-Detached, T = Terraced, F = Flats/Maisonettes, O = Other
Old/New	Indicates the age of the property and applies to all price paid transactions, residential and non-residential. Y = a newly built property, N = an established residential building
Duration	Relates to the tenure: F = Freehold, L= Leasehold etc. Note that HM Land Registry does not record leases of 7 years or less in the Price Paid Dataset.
PAON	Primary Addressable Object Name. Typically the house number or name.
SAON	Secondary Addressable Object Name. Where a property has been divided into separate units (for example, flats), the PAON (above) will identify the building and a SAON will be specified that identifies the separate unit/flat.

Continued on next page

Table 3 – continued from previous page

First column	**Second column**
Street	
Locality	
Town/City	
District	
County	
PPD Category Type	Indicates the type of Price Paid transaction. A = Standard Price Paid entry, includes single residential property sold for value. B = Additional Price Paid entry including transfers under a power of sale/repossessions, buy-to-lets (where they can be identified by a Mortgage) and transfers to non-private individuals.Note that category B does not separately identify the transaction types stated. HM Land Registry has been collecting information on Category A transactions from January 1995. Category B transactions were identified from October 2013.

A.2 EPC attribute descriptions

Format:
Attribute name COLUMN_LABEL
Description

LMK key LMK_KEY
Individual lodgement identifier. Guaranteed to be unique and can be used to identify a certificate in the downloads and the API.

Address 1 ADDRESS1
First line of the address

Address 2 ADDRESS2
Second line of the address

Address 3 ADDRESS3
Third line of the address

Postcode POSTCODE
The postcode of the property

Building reference number BUILDING_REFERENCE_NUMBER
Unique identifier for the property. Generated by the Energy Performance of Buildings Registers within 24 hours to minimise delays to the EPC lodgement process and for reference numbers to be created for building units in larger premises. Not the same as commonly used unique property reference numbers.

Current energy rating CURRENT_ENERGY_RATING
Current energy rating converted into a linear ’A to G’ rating (where A is the most energy efficient and G is the least energy efficient)

Potential energy rating POTENTIAL_ENERGY_RATING
Estimated potential energy rating converted into a linear ’A to G’ rating (where A is the most energy efficient and G is the least energy efficient)

Current energy efficiency CURRENT_ENERGY_EFFICIENCY

Based on cost of energy, i.e. energy required for space heating, water heating and lighting [in kWh/year] multiplied by fuel costs. (£/m/year where cost is derived from kWh).

Potential energy efficiency POTENTIAL_ENERGY_EFFICIENCY

The potential energy efficiency rating of the property.

Property type PROPERTY_TYPE

Describes the type of property such as House, Flat, Maisonette etc. This is the type differentiator for dwellings.

Built form BUILT_FORM

The building type of the Property e.g. Detached, Semi-Detached, Terrace etc. Together with the Property Type, the Build Form produces a structured description of the property

Inspection date INSPECTION_DATE

The date that the inspection was actually carried out by the energy assessor

Local authority LOCAL_AUTHORITY

Office for National Statistics (ONS) code. Local authority area in which the building is located.

Constituency CONSTITUENCY

Office for National Statistics (ONS) code. Parliamentary constituency in which the building is located.

County COUNTY County in which the building is located (where applicable)

Lodgement date LODGEMENT_DATE

Date lodged on the Energy Performance of Buildings Register

Transaction type TRANSACTION_TYPE

Type of transaction that triggered EPC. For example, one of: marketed sale; non-marketed sale; new-dwelling; rental; not sale or rental; assessment for Green Deal; following Green Deal; FIT application; none of the above; RHI application; ECO assessment. Where the reason for the assessment is unknown by the energy assessor the transaction type will be recorded as 'none of the above'. Transaction types may be changed over time.

Environment impact current ENVIRONMENT_IMPACT_CURRENT

The Environmental Impact Rating. A measure of the property's current impact on the environment in terms of carbon dioxide (CO) emissions. The higher the rating the lower the CO emissions. (CO emissions in tonnes / year)

Environment impact potential ENVIRONMENT_IMPACT_POTENTIAL

The potential Environmental Impact Rating. A measure of the property's potential impact on the environment in terms of carbon dioxide (CO) emissions after improvements have been carried out. The higher the rating the lower the CO emissions. (CO emissions in tonnes / year)

Energy consumption current ENERGY_CONSUMPTION_CURRENT

Current estimated total energy consumption for the property in a 12 month period (kWh/m2). Displayed on EPC as the current primary energy use per square metre of floor area.

Energy consumption potential ENERGY_CONSUMPTION_POTENTIAL

Estimated potential total energy consumption for the Property in a 12 month period. Value is Kilowatt Hours per Square Metre (kWh/m)

Co emissions current CO2_EMISSIONS_CURRENT

CO emissions per year in tonnes/year.

Co emiss curr per floor area CO2_EMISS_CURR_PER_FLOOR_AREA

CO emissions per square metre floor area per year in kg/m

Co emissions potential CO2_EMISSIONS_POTENTIAL

Estimated value in Tonnes per Year of the total CO emissions produced by the Property in 12 month period.

Lighting cost current LIGHTING_COST_CURRENT

GBP. Current estimated annual energy costs for lighting the property.

Lighting cost potential LIGHTING_COST_POTENTIAL

GBP. Potential estimated annual energy costs for lighting the property after improvements have been made.

Heating cost current HEATING_COST_CURRENT

GBP. Current estimated annual energy costs for heating the property.

Heating cost potential HEATING_COST_POTENTIAL

GBP. Potential annual energy costs for lighting the property after improvements have been made.

Hot water cost current HOT_WATER_COST_CURRENT

GBP. Current estimated annual energy costs for hot water

Hot water cost potential HOT_WATER_COST_POTENTIAL

GBP. Potential estimated annual energy costs for hot water after improvements have been made.

Total floor area TOTAL_FLOOR_AREA

The total useful floor area is the total of all enclosed spaces measured to the internal face of the external walls, i.e. the gross floor area as measured in accordance with the guidance issued from time to time by the Royal Institute of Chartered Surveyors or by a body replacing that institution. (m)

Energy tariff ENERGY_TARIFF

Type of electricity tariff for the property, e.g. single.

Mains gas flag MAINS_GAS_FLAG

Whether mains gas is available. Yes means that there is a gas meter or a gas-burning appliance in the dwelling. A closed-off gas pipe does not count.

Floor level FLOOR_LEVEL

Flats and maisonettes only. Floor level relative to the lowest level of the property (0 for ground floor). If there is a basement, the basement is level 0 and the other floors are from 1 upwards

Flat top storey FLAT_TOP_STOREY

Whether the flat is on the top storey

Flat storey count FLAT_STOREY_COUNT

The number of storeys in the apartment block.

Main heating controls MAIN_HEATING_CONTROLS Type of main heating controls. Includes both main heating syste ms if there are two.

Multi glaze proportion MULTI_GLAZE_PROPORTION

The estimated banded range (e.g. 0% - 10%) of the total glazed area of the Property that is multiple glazed.

Glazed type GLAZED_TYPE

The type of glazing. From British Fenestration Rating Council or manufacturer declaration, one of; single; double; triple.

Glazed area GLAZED_AREA

Ranged estimate of the total glazed area of the Habitable Area.

Extension count EXTENSION_COUNT

The number of extensions added to the property. Between 0 and 4.

Number habitable rooms NUMBER_HABITABLE_ROOMS

Habitable rooms include any living room, sitting room, dining room, bedroom, study and similar; and also a non-separated conservatory. A kitchen/diner having a discrete seating area (with space for a table and four chairs) also counts as a habitable room. A non-separated conservatory adds to the habitable room count if it has an internal quality door between it and the dwelling. Excluded from the room count are any room used solely as a kitchen, utility room, bathroom, cloakroom, en-suite accommodation and similar and any hallway, stairs or landing; and also any room not having a window.

Number heated rooms NUMBER_HEATED_ROOMS

The number of heated rooms in the property if more than half of the habitable rooms are not heated.

Low energy lighting LOW_ENERGY_LIGHTING

The percentage of low energy lighting present in the property as a percentage of the total fixed lights in the property. 0% indicates that no low-energy lighting is present.

Number open fireplaces NUMBER_OPEN_FIREPLACES

The number of Open Fireplaces in the Property. An Open Fireplace is a fireplace that still allows air to pass between the inside of the Property and the outside.

Hotwater description HOTWATER_DESCRIPTION

Overall description of the property feature

Hot water energy eff HOT_WATER_ENERGY_EFF

Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Hot water env eff HOT_WATER_ENV_EFF

Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Floor description FLOOR_DESCRIPTION

Overall description of the property feature

Floor energy eff FLOOR_ENERGY_EFF

Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Floor env eff FLOOR_ENV_EFF

Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Windows description WINDOWS_DESCRIPTION

Overall description of the property feature

Windows energy eff WINDOWS_ENERGY_EFF

Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Windows env eff WINDOWS_ENV_EFF

WINDOWS. Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Walls description WALLS_DESCRIPTION

Overall description of the property feature

Walls energy eff WALLS_ENERGY_EFF

Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Walls env eff WALLS_ENV_EFF

Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Overall description of the property feature

Sheating energy eff SHEATING_ENERGY_EFF Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Sheating env eff SHEATING_ENV_EFF Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Roof description ROOF_DESCRIPTION Overall description of the property feature

Roof energy eff ROOF_ENERGY_EFF Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Roof env eff ROOF_ENV_EFF Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Mainheat description MAINHEAT_DESCRIPTION Overall description of the property feature

Mainheat energy eff MAINHEAT_ENERGY_EFF Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Mainheat env eff MAINHEAT_ENV_EFF Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Mainheatcont description MAINHEATCONT_DESCRIPTION

Overall description of the property feature

Mainheatc energy eff MAINHEATC_ENERGY_EFF

Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Mainheatc env eff MAINHEATC_ENV_EFF

Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Lighting description LIGHTING_DESCRIPTION

Overall description of property feature. Total number of fixed lighting outlets and total number of low-energy fixed lighting outlets

Lighting energy eff LIGHTING_ENERGY_EFF

Energy efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Lighting env eff LIGHTING_ENV_EFF

Environmental efficiency rating. One of: very good; good; average; poor; very poor. On actual energy certificate shown as one to five star rating.

Main fuel MAIN_FUEL

The type of fuel used to power the central heating e.g. Gas, Electricity

Wind turbine count WIND_TURBINE_COUNT

Number of wind turbines; 0 if none.

Heat loss corridoor HEAT_LOSS_CORRIDOOR

Flats and maisonettes only. Indicates that the flat contains a corridor through which heat is lost. Heat loss corridor, one of: no corridor; heated corridor; unheated corridor

Unheated corridor length UNHEATED_CORRIDOR_LENGTH

The total length of unheated corridor in the flat. Only populated if flat or maisonette contains unheated corridor. If unheated corridor, length of sheltered wall (m).

Floor height FLOOR_HEIGHT

Average height of the storey in metres.

Photo supply PHOTO_SUPPLY

Percentage of photovoltaic area as a percentage of total roof area. 0% indicates that a Photovoltaic Supply is not present in the property.

Solar water heating flag SOLAR_WATER_HEATING_FLAG

Indicates whether the heating in the Property is solar powered.

Mechanical ventilation MECHANICAL_VENTILATION

Identifies the type of mechanical ventilation the property has. This is required for the RdSAP calculation.

Address ADDRESS

Field containing the concatenation of address1, address2 and address3. Note that post code is recorded separately.

Local authority name LOCAL_AUTHORITY_LABEL

The name of the local authority area in which the building is located. This field is for additional information only and should not be relied upon: please refer to the Local Authority ONS Code.

Constituency name CONSTITUENCY_LABEL

The name of the parliamentary constituency in which the building is located.

This field is for additional information only and should not be relied upon: please refer to the Constituency ONS Code.

Post town POSTTOWN

The post town of the property

Construction age band CONSTRUCTION_AGE_BAND

Age band when building part constructed. England Wales only. One of: before 1900; 1900-1929; 1930-1949; 1950-1966; 1967-1975; 1976-1982; 1983-1990; 1991-1995; 1996-2002; 2003-2006; 2007-2011; 2012 onwards.

Lodgement datetime LODGEMENT_DATETIME

Date and time lodged on the Energy Performance of Buildings Register.

Tenure TENURE

Describes the tenure type of the property. One of: Owner-occupied; Rented (social); Rented (private).

Fixed lighting outlets count FIXED_LIGHTING_OUTLETS_COUNT

The number of fixed lighting outlets.

Low energy fixed lighting outlets count LOW_ENERGY_FIXED_LIGHT_COUNT

The number of low-energy fixed lighting outlets.

A.3 Index of Multiple Deprivation domain definitions

Index of Multiple Deprivation
The Index of Multiple Deprivation 2019 combines information from the seven domains to produce an overall relative measure of deprivation. The domains are combined using the following weights: Income Deprivation (22.5%), Employment Deprivation (22.5%), Education, Skills and Training Deprivation (13.5%), Health Deprivation and Disability (13.5%), Crime (9.3%), Barriers to Housing and Services (9.3%), Living Environment Deprivation (9.3%).

Income Deprivation Domain
The Income Deprivation Domain measures the proportion of the population experiencing deprivation relating to low income. The definition of low income used includes both those people that are out-of-work, and those that are in work but who have low earnings(and who satisfy the respective means tests).

Employment Deprivation Domain
The Employment Deprivation Domain measures the proportion of the working age population in an area involuntarily excluded from the labour market. This includes people who would like to work but are unable to do so due to unemployment, sickness or disability, or caring responsibilities.

Education, Skills and Training Deprivation Domain The Education, Skills and Training Deprivation Domain measures the lack of attainment and skills in the local population. The indicators fall into two sub-domains: one relating to children and young people and one relating to adult skills.

Health Deprivation and Disability Domain
The Health Deprivation and Disability Domain measures the risk of premature death and the impairment of quality of life through poor physical or mental health. The domain measures morbidity, disability and premature mortality but not aspects of behaviour or environment that may be predictive of future health deprivation.

Crime Domain
The Crime Domain measures the risk of personal and material victimisation at

local level. 24The English Indices of Deprivation 2019 -Statistical Release

Barriers to Housing and Services Domain

The Barriers to Housing and Services Domain measures the physical and financial accessibility of housing and local services. The indicators fall into two sub-domains: ‘geographical barriers’, which relate to the physical proximity of local services, and ‘wider barriers’ which includes issues relating to access to housing such as affordability and homelessness.

Living Environment Deprivation Domain

The Living Environment Deprivation Domain measures the quality of the local environment. The indicators fall into two sub-domains. The ‘indoors’ living environment measures the quality of housing; while the ‘outdoors’ living environment contains measures of air quality and road traffic accidents.

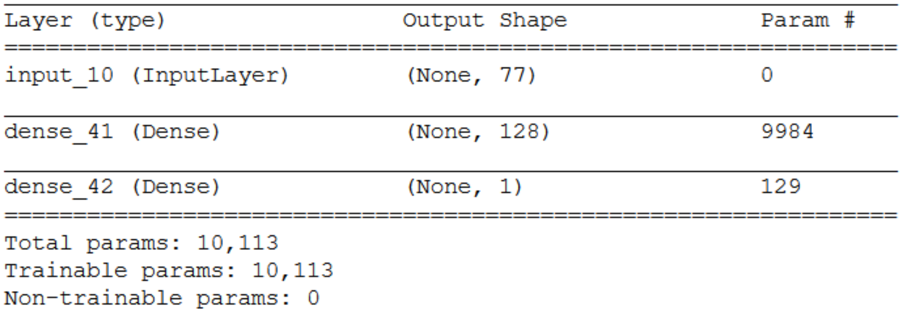
Income Deprivation Affecting Children Index

The Income Deprivation Affecting Children Index (IDACI) measures the proportion of all children aged 0 to 15 living in income deprived families. Family is used here to indicate a ‘benefit unit’, that is the claimant, any partner and any dependent children for whom Child Benefit is received. This is one of two supplementary indices and is a sub-set of the Income Deprivation Domain.

Income Deprivation Affecting Older People Index

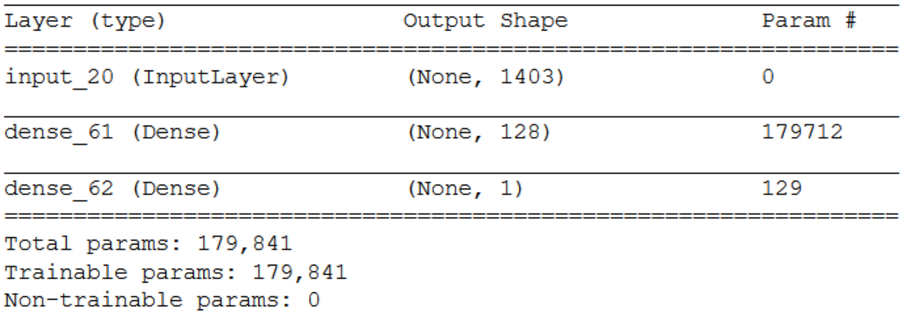
The Income Deprivation Affecting Older People Index (IDAOPI) measures the proportion of all those aged 60 or over who experience income deprivation. This is one of two supplementary indices and is a sub-set of the Income Deprivation Domain

B Keras combined model summaries



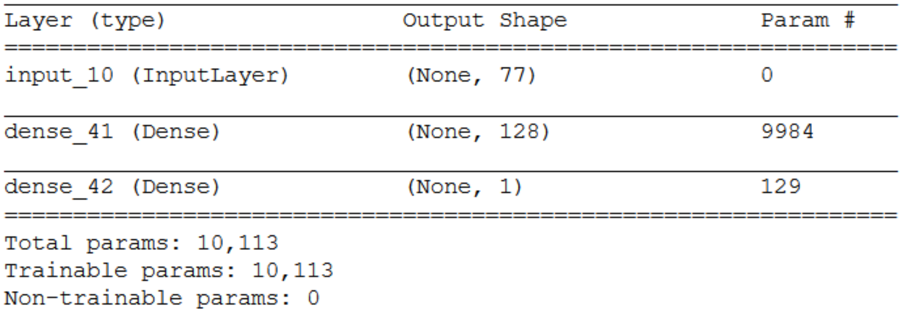
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_10 (InputLayer)        (None, 77)                0
_________________________________________________________________
dense_41 (Dense)             (None, 128)               9984
_________________________________________________________________
dense_42 (Dense)             (None, 1)                 129
=================================================================
Total params: 10,113
Trainable params: 10,113
Non-trainable params: 0
```

Figure 18: Architecture A with Basic CNN features



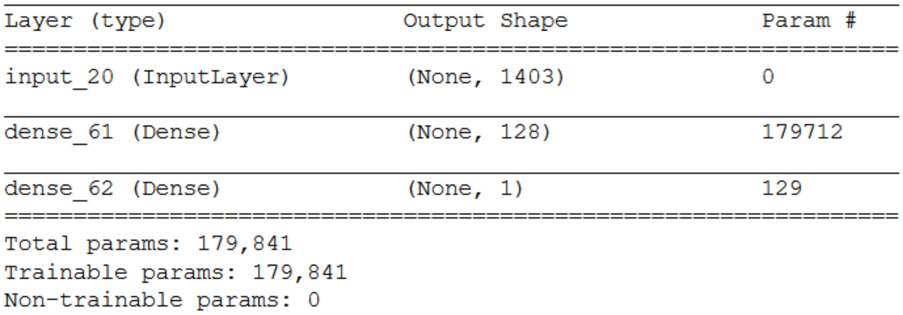
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_20 (InputLayer)        (None, 1403)              0
_________________________________________________________________
dense_61 (Dense)             (None, 128)               179712
_________________________________________________________________
dense_62 (Dense)             (None, 1)                 129
=================================================================
Total params: 179,841
Trainable params: 179,841
Non-trainable params: 0
```

Figure 19: Architecture A with ResNet50 features



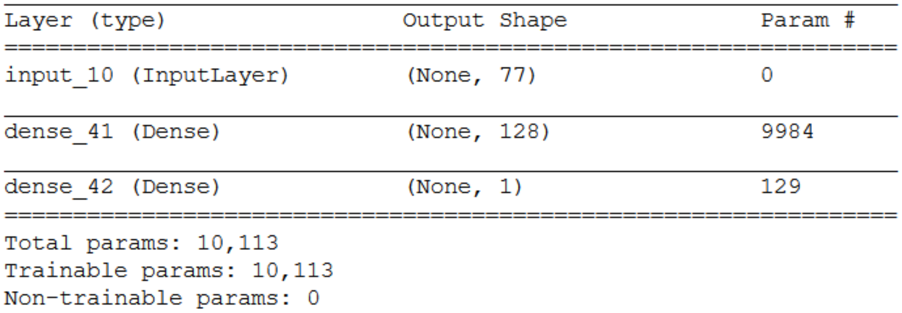
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_10 (InputLayer)        (None, 77)                0
_________________________________________________________________
dense_41 (Dense)             (None, 128)               9984
_________________________________________________________________
dense_42 (Dense)             (None, 1)                 129
=================================================================
Total params: 10,113
Trainable params: 10,113
Non-trainable params: 0
```

Figure 20: Architecture B with Basic CNN features



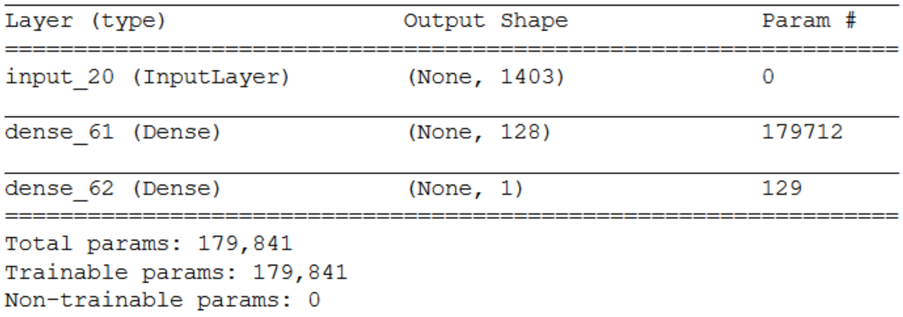
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_20 (InputLayer)        (None, 1403)              0
_________________________________________________________________
dense_61 (Dense)             (None, 128)               179712
_________________________________________________________________
dense_62 (Dense)             (None, 1)                 129
=================================================================
Total params: 179,841
Trainable params: 179,841
Non-trainable params: 0
```

Figure 21: Architecture B with ResNet50 features



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_10 (InputLayer)        (None, 77)                0
_________________________________________________________________
dense_41 (Dense)             (None, 128)               9984
_________________________________________________________________
dense_42 (Dense)             (None, 1)                 129
=================================================================
Total params: 10,113
Trainable params: 10,113
Non-trainable params: 0
```

Figure 22: Architecture C with Basic CNN features



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_20 (InputLayer)        (None, 1403)              0         
_________________________________________________________________
dense_61 (Dense)             (None, 128)               179712    
_________________________________________________________________
dense_62 (Dense)             (None, 1)                 129       
=================================================================
Total params: 179,841
Trainable params: 179,841
Non-trainable params: 0
```

Figure 23: Architecture C with ResNet50 features