

# Automatic Assignment of Geospatial Coordinates to Historical Photos

Nuno Yatin Manojé Ramanlal

Instituto Superior Técnico

## Abstract

This document describes a machine-learning based approach to automatically geocode historical photos, leveraging convolutional neural network architectures. These networks require vast amounts of training data, and while social media and photo sharing platforms correspond to two of the most significant sources of data for performing supervised learning, there are not large datasets of old geocoded photos made available, which makes the task of geocoding old historical photos remain not well-covered. This work introduces an end-to-end network, combining a convolutional neural network model based on the ResNet architecture for automated geo-referencing, with another fully-convolutional network, also based on the ResNet architecture, to perform transformations over old photos, in an attempt to resemble the modern ones. This way, the geocoding network can be pre-trained with modern photos retrieved from the Flickr dataset. There are reported several geocoding experiments over different network settings, leveraging existing collections of geo-referenced historical photos. Experiment's results show that the proposed network is more efficient geocoding historical photos in comparison to other evaluated convolutional neural networks that do not use the coloring component, which highlights the potential of this approach.

**Keywords**— Geocoding Historical Photos, Deep Learning for Analyzing Photos, Convolutional Neural Architectures

## 1 Introduction

Historical photos can carry rich information that may be very useful for a variety of different applications. Some examples relate to landscape change and evolution, land planning, land cover mapping, or in order to better understanding cultural heritage. With the emergence of the Web 2.0, - a term commonly used to refer to the current state of online technology - followed by the constant development of social media platforms based on photo sharing, users are now able to share historical photos through such Web platforms<sup>1</sup>. Considering the period at which these old photos were taken, GPS-enabled devices or cameras were not yet available, and therefore accurate location information of those is not available. This issue has two potential consequences: 1) photos are shared without any location information; or 2) they are manually assigned to a location by those who are uploading them, with potential errors being made depending on their knowledge and geographic cognition.

Geocoding photos, particularly old ones, can be relevant in the tasks of historical reconstructions and understanding changes over time. The majority of the introduced approaches based on machine learning techniques perform the geocoding by processing images, specifically analyzing the data contained in those, which increases the efforts to make progress in this field. Alongside machine learning approaches, many devices became capable of performing automated geocoding leveraging built-in GPS features. In the absence of GPS signal, or if the data is not ready to be processed by such devices, manual geocoding can be a long but possible task.

Previous work has already been conducted, aiming to extract and explore the potential that this visual data provides. However, there are still some tasks that automatic content-based photo geocoding has not yet covered properly. One of those concerns with geocoding old photos automatically. Determining the geolocation of photos can solve two main issues: 1) it allows to correct any mistakes done by users while geocoding photos manually; and 2) it allows to automatically geocode historical photos with reasonable accuracy, which can be challenging for human beings. I propose to tackle the task of geocoding historical photos through supervised learning, using the Yahoo Flickr Creative Commons 100 Million dataset, introduced by Thomee et al. [1], to train the proposed models. This task will also require performing image transformations, specifically image colorization, based on the recent deep learning algorithms. The final model consists in a network that will first perform the coloring task, as an attempt to adapt old photos to the characteristics of recent ones, and then perform the geocoding task over colored historical photos using the models pre-trained with the Flickr dataset.

The rest of the document is organized as follows: Section 2 introduces some concepts related to the proposed task and describes briefly some related work done in the field of automated photo geocoding. Section 3 presents the proposed deep neural network architecture. Section 4 reports the results of the performed experiments, and finally, Section 5 presents the conclusions and future work that may be interesting to do in the field.

---

<sup>1</sup><http://www.pinterest.pt>, <http://www.flickr.com>

## 2 Concepts and Related Work

This section is divided into two other subsections. Section 2.1 reports fundamental concepts of machine learning applied in this work, such as Convolutional Neural Networks (CNNs). Section 2.2 presents some previous studies conducted in the field of geocoding photos, reporting different successful methods to do so.

### 2.1 Deep Learning for Image Processing

Neural networks are computational artifacts that channel information through a series of mathematical operations, with the general purpose of accurately mapping inputs to target values. Mathematically, neural networks can be seen as nested composite functions, whose parameters can be trained directly to minimize a given loss function computed over the outputs and the expected results. This is achieved through a training procedure known as back-propagation, in combination with gradient descent optimization of the parameters.

In the simplest case, a single-node neural network computes a single output from multiple real-valued inputs by forming a linear combination according to input weights, and then putting the output through some activation function. Mathematically, this can be written as shown in Equation 1, where  $y$  refers to the returned prediction,  $x = \langle x_1, \dots, x_n \rangle$  is the vector of input features,  $w$  denotes the vector of weights,  $b$  is a bias term, and  $\varphi(\cdot)$  is an activation function (e.g., a logistic sigmoid, a hyperbolic tangent, or the nowadays common rectified linear unit activation).

$$y = \varphi \left( \sum_{i=1}^n w_i \times x_i + b \right) = \varphi (w^T \cdot x + b) \quad (1)$$

Although a single neural network node has a limited mapping ability, the same idea can be used at the main building block of more complex models. For instance, a Multi-Layer Perceptron (MLP) consists of a set of nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of nodes. The input signal propagates through the network layer-by-layer, until it reaches the output node(s). In a feed-forward network with a single hidden layer, the corresponding computations can be written as shown in Equation 2, and the generalization to more hidden layers would be simple.

$$y = \varphi (B \times \varphi' (A \times x + a) + b) \quad (2)$$

In the previous equation,  $x$  is a vector of inputs and  $y$  a vector of outputs. The matrix  $A$  represents the weights of the first layer and  $a$  is the bias vector of the first layer, while  $B$  and  $b$  are, respectively, the weight matrix and the bias vector of the second layer. The functions  $\varphi'$  and  $\varphi$  both denote the activation functions respectively associated to nodes in the hidden layer, and in the output layer.

Training the neural network corresponds to adapting all the weights and biases to their optimal values, given a training set of inputs  $x$  and the corresponding outputs  $y$ . This problem can be solved with the back-propagation algorithm, which consists of two steps [2]. In a forward pass, the predicted outputs corresponding to the given inputs are evaluated. In a backward pass, the error calculated from the predicted outputs in the output layer is propagated backwards throughout the layers, updating the layer's weight values.

An important limitation of MLPs is related to the computational complexity required to process image data (i.e., the number of parameters associated to processing images, assuming the network inputs correspond to the individual pixel values, would be very high). CNNs are alternative architectures that address this issue, by having the neurons within any given layer only connecting to a small region of the layer preceding it, and by using common parameters for processing all these small regions. In more detail, CNNs are comprised of three types of layers, namely convolutional layers and pooling layers (i.e., layers where neurons are organized into three dimensions of height, width, and depth), combined with fully-connected layers similar to those in MLPs, that are responsible for generating the final representations and the outputs. The basic functionality of a CNN architecture can be broken down as follows:

1. The input to the network holds the pixel values of an image, corresponding to a 2D matrix (i.e., one single channel encoding pixel intensities) or a 3D tensor (i.e., an image with different color channels).
2. The convolutional layer determines the output of neurons connected to local regions of the input, through the calculation of the scalar product between the layer's weights and the region connected to the input volume, followed by an activation function. The depth of the output produced by a convolutional layer corresponds to a number of filters. Each filter is convolved across the spatial dimensionality of the input, producing a 2D activation map. These maps are stacked along the depth dimension to form the full output volume from the convolutional layer.
3. The pooling layer will then perform downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation. In most CNNs, these come in the form of max-pooling or average-pooling layers, with filters of dimensionality  $2 \times 2$ , applied with a stride of 2 along the spatial dimensions of the input.
4. The fully-connected layers, similar to those found on MLPs, will attempt to produce the desired outputs from the previous activations.

	Street	City	Region	Country	Continent
Threshold (Km)	1	25	200	750	2500
Human assignment			3.8	13.9	39.3
Im2GPS [3]		12.0	15.0	23.0	47.0
PlaNet [4]	08.4	24.5	37.6	53.6	71.3
Classification model with 7011 classes	06.8	21.9	34.6	49.4	63.7
Regression model	12.2	33.3	44.3	57.4	71.3
Regression model trained with 28M photos	14.4	33.3	47.7	61.6	73.4

Table 1: Accuracy results in the geocoding experiment reported by Vo et al. [6].

## 2.2 Automatically Geocoding Photos

Geocoding tasks are often formulated as either regression or classification problems. In regression tasks, one must expect to be presented to a pair of inferred geographic coordinates as the output, while in classification tasks the output is represented by a probabilistic distribution over a finite set of possible locations. In two studies of applying deep learning techniques as geolocation methods, both Hays et al. [3] and Weyand et al. [4] introduced state-of-art approaches in the task of geolocating arbitrary photos from across the world.

The approach followed by Hays et al. [3], called Im2GPS, was the first to be able to extract geographic information from a single image. The method consists of using data-driven scene matching between a dataset gathering 6 million of both keywords and GPS coordinates labeled images from the Flickr online collection and a photo from an evaluation test set (i.e., a query photo). For each image of the dataset and the given query image, there are measured the distances between the following features: color image space, color and texton histograms, statistics of straight lines in images, gist descriptor, and finally geometric class probabilities for image regions. Once the measurements are completed, the estimated location can be either based in the first nearest neighbor, or represented by a probability map over the entire globe representing the  $k$  nearest neighbors.

The approach followed by Weyand et al. [4], which resulted in a model called PlaNet, consists in subdividing the Earth’s surface into geographic cells to define the target classes. For this classification task, the CNN used is based on the Inception architecture, introduced by Szegedy et al. [5], outputting a one-hot vector encoding the cell corresponding to the query image localization. To measure the model accuracy, and since the query photos are assigned to geographic cells, the localization error is measured calculating the distance between the center of the predicted cell and the actual location of the photo.

Vo et al. [6] combined both these state-of-the-art approaches described above, adopting the retrieval approach of Im2GPS while learning deep features as in PlaNet. The authors also conducted several experiments in which they used different image classification and retrieval methods. In both cases, the CNN architecture was based on the VGG-16 network, introduced by Simonyan et al. [7]. In the classification task, as in PlaNet, the authors repeatedly divided the Earth’s surface into cells considering the number of images per cell and the physical area. The experiments were performed over 6 different partitions, and also using the 6 partitions simultaneously, leading to the conclusion that there is a trade-off between the accuracy at coarse and fine level, considering the partition numbers. To perform geolocalization as a regression task, the authors learned a representation to compare a query image with a database composed of geocoded photos, looking for resemblances between images.

Table 1 reports the performance of all three described approaches on the Im2GPS test set. PlaNet obtains far more accurate results than Im2GPS, and it also outperforms the *Classification model (7011 classes)*, which corresponds to the classification network trained with 7011 classes by Vo et al. [6]. The *Regression model* corresponds to the kNN kernel density estimation retrieval, which was also introduced in the last study introduced in this chapter. The last row in the table refers to the same retrieval approach but trained with a larger dataset (Yahoo Flickr Creative Commons 100 Million). This last model obtains the best results in all metrics, showing that the accuracy improve when using a larger training set.

## 3 The Proposed Deep Neural Network Architecture

This section describes the neural network components used in the approach introduced to address the task of geocoding historical photos. Section 3.1 details the ResNet network structure, which builds on the concept of residual network blocks. The model presented high accuracy results in case studies and succeeded in several machine learning competitions. For instance, an ensemble of residual networks achieved 3.57% error in the ImageNet test set, allowing the authors to win the *1st* place on the ILSVRC 2015 classification task<sup>2</sup>. Section 3.2 reports the approach and the models used to build the image transformation component and the geocoding component.

<sup>2</sup><http://image-net.org/challenges/LSVRC/2015/>

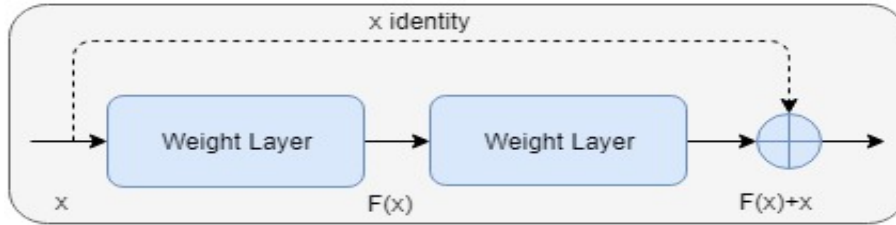


Figure 1: Residual learning: a building block.

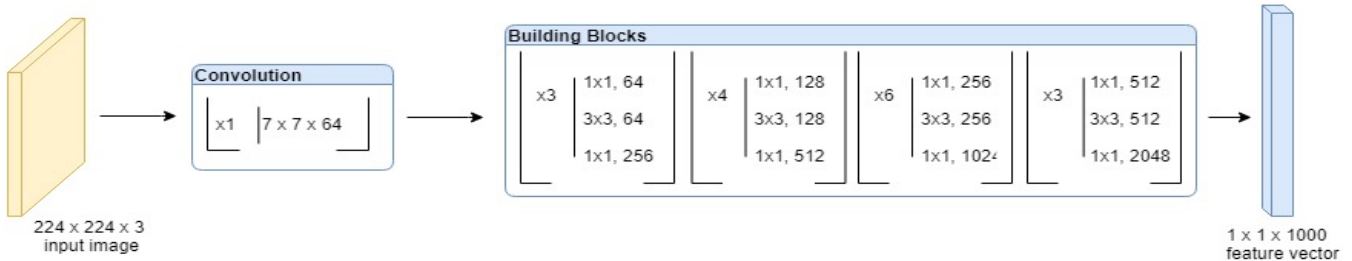


Figure 2: ResNet50 architecture.

### 3.1 The ResNet Architecture for Image Classification

The success of visual recognition tasks leveraging machine learning techniques is highly dependent on the depth of the CNNs. Studies like the one performed by Simonyan et al. [7] have demonstrated that deep networks perform well in comparison to their shallower counterparts. However, Simonyan et al. [7], in the same case study, concluded that the accuracy gets saturated as the networks get deeper. In some cases the accuracy might also start degrading. Motivated by this, He et al. [8] presented a residual learning framework to address the issue. The approach consisted in learning the difference  $F(x) = H(x) - x$  between the desired underlying mapping  $H(x)$  to be fit by a set of stacked layers, instead of learning a mapping directly between  $x$  and  $H(x)$ . Thus, the desired mapping  $H(x)$  becomes  $F(x) + x$ .

Based on these principles, the authors introduced a residual network (ResNet) block, which is illustrated in Figure 1. Each block is composed by a set of layers, and by a shortcut connection in charge of performing identity mapping and making element-wise additions on feature maps. This work leverages the 50-layer ResNet, named ResNet50, whose architecture is illustrated in Figure 2. The network is based on residual blocks of 3 layers, instead of the 2 layers residual blocks used in ResNet34, whose architecture comprises 34 layers.

In a case study, the authors took two plain networks with 18 and 34 layers respectively, and two other residual networks, also with 18 and 34 layers each. All four models were trained from scratch on 1.28 million images and evaluated on the ImageNet validation set. Both ResNet models performed better in comparison to the plain networks, with the ResNet with 34 layers performing better than all the other architectures, with a top-1 error of 25.03%.

### 3.2 A Deep Neural Network for Geocoding Historical Photos

The proposed network to geocode historical photos is a result of combining two different CNNs: a first network trained to color grayscale images, and a second one trained to perform geocoding tasks. Given a  $224 \times 224 \times 3$  input photo, a deep network is first used to color it, and the output is then fed to a geocoding network which is expecting a colored photo. The result is a sequential convolution workflow, capable of performing geospatial predictions over historical query photos. The following subsections describe the networks trained to perform the above mentioned tasks, i.e., color grayscale photos and geocode those images. Specifically, Section 3.2.1 details the image transformation network, and Section 3.2.2 reports the networks used to geocode historical photos.

#### 3.2.1 Coloring Historical Photos

To perform the task of coloring historical photos, a model capable of producing a mapping between the luminance values of a grayscale image and color values, i.e., color-wise transformations, was exploited. The model's architecture, illustrated in Figure 3, was made available by Majumdar in his GitHub repository<sup>3</sup>. It is fully based on a work introduced by Baldassarre et al. [9], in which a CNN was trained from scratch and combined with a pre-trained Inception-Resnet-v2 network, introduced by Szegedy et al. [10], in order to extract high-level features. Each input image is processed by the network's four components: the encoder, the feature extractor, the fusion layer, and the decoder.

At both training and inference time, the model is fed with a  $224 \times 224 \times 3$  input image. The encoder starts by processing the input image just considering the luminance channel of it. This component applies several convolutional operations, outputting a reduced size image representation with 256 new channels. The feature extractor used in Majumdar's architecture, the MobileNet network, introduced by Howard et al. [11], was replaced with the ResNet50

<sup>3</sup><http://github.com/titu1994/keras-mobile-colorizer>

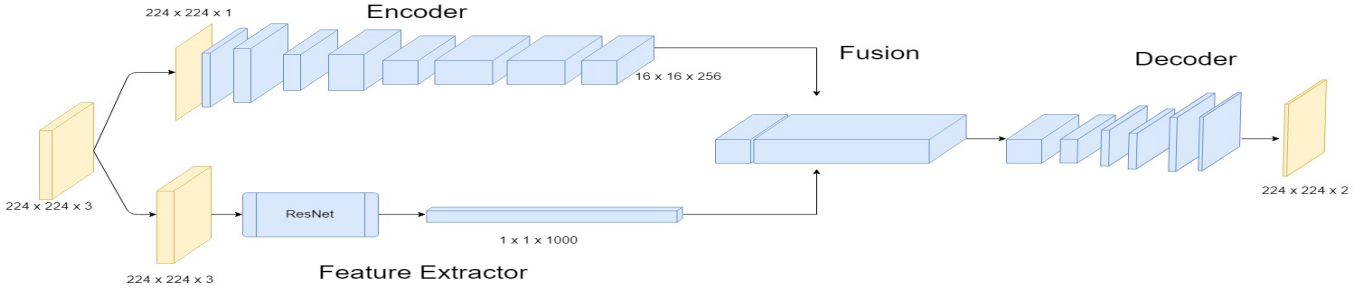


Figure 3: Illustration of the Colorizer Network Architecture.

network, which is fed with the same input image as the encoder, but this time considering the 3 channels of the CIEL\*a\*b\* color space. The fusion layer joins both the encoder and the feature extractor outputs, and the decoder applies another set of convolution operations together with upsampling operations to build a two channels colored version of the input, projecting the inferred colors.

This network was trained with 70000 randomly chosen pictures from Yahoo Flickr Creative Commons 100 Million dataset, using Adam optimizer with an initial learning rate of 0.0001, and employing the mean square error between the estimated a\*b\* values and their real values as the loss function. Figure 4 presents the results of testing the model on photos extracted from an article published in the Guardian journal<sup>4</sup>, which illustrates the transformations of sites across the San Francisco Bay Area. Notice that the first and third rows represent the same place, but in different time periods. The modern pictures were converted in grayscale and afterwards colored by the introduced network. To quantitatively evaluate the performance of this component, the average Root Mean Square Error (RMSE) between the original photos and the colored versions were measured, similarly to Deshpande et al. [12]. The average RMSE value between the original and the colored version of the pictures was 18.540, whereas between the original and the grayscale version of the same set of photos was 18.608. The results show that the trained model is in fact coloring the photos, however, the difference between these two values is not as big as it would be preferable.

### 3.2.2 Geocoding Photos Combining Classification and Regression Outputs

To geocode historical photos, it was proposed a model capable of generating two different types of outputs, i.e., outputs corresponding to regression and classification tasks. The network was trained with the corresponding losses for each task, back-propagating the error values throw out the network as an attempt to learn and update the parameters in a more precise way and improve the accuracy. Leveraging the ResNet50 network described in Section 3.1, the last fully-connected layer from the model was removed, appending to it layers corresponding to the regression and the classifications components. It was also applied global average pooling, and the model’s parameters were initialized with the ImageNet weights made available at keras library<sup>5</sup>. Figure 5 illustrates the described network’s architecture.

Considering the regressions task, the network was trained to output a pair of coordinates corresponding to the predicted geolocation of the query photo. To do so, the fully-connected layer was replaced with a dense layer using a sigmoid activation, and with an output dimensionality corresponding to a pair of geospatial coordinates. The loss function used to train the network parameters was a special case of Vincenty’s formulae, introduced by Vincenty et al. [13], which assumes an ellipsoid with equal major and minor axes, and it is defined as follows:

$$\arctan \frac{\sqrt{(\cos \phi_2 \cdot \sin (\Delta \lambda))^2 + (\cos \phi_1 \cdot \sin \phi_2 - \sin \phi_1 \cdot \cos \phi_2 \cdot \cos (\Delta \lambda))^2}}{\sin \phi_1 \cdot \sin \phi_2 + \cos \phi_1 \cdot \cos \phi_2 \cdot \cos (\Delta \lambda)} \quad (3)$$

In the previous equation,  $\phi_1$  and  $\phi_2$  correspond to the geographic latitude values in radians, and  $\Delta \lambda$  represent the absolute difference between the geographic longitude values, which are also in radians. In this specific task, the loss function measures the distance between the inferred and real coordinates in each training instance.

To get the output corresponding to the classification task, another dense layer was added, this time using softmax as the activation function, which returns a tensor representing the probabilistic distribution over the target classes. These target classes correspond to non-overlapping cells representing different regions of the Earth’s surface. At training time, the model learned to make classification tasks using categorical crossentropy as the loss function. The division of this spheroid surface was done leveraging the Healpy<sup>6</sup> Python wrapper for Hierarchical Equal Area isoLatitude Pixelation (HEALPix)<sup>7</sup>, introduced by Grski et al. [14]. HEALPix is a pixelation technique that subdivides a spherical surface in which, at a given resolution, the areas of all pixels are identical. In this method, a sphere is hierarchically tessellated into curvilinear quadrilaterals. The resolution of the tessellation increases by the division of each pixel into four new ones, being the lowest resolution partition comprised of 12 pixels.

Notice that these architecture’s descriptions correspond to the two different parts attached to the ResNet50 network after removing the last fully-convolutional layer, and that there were not made any additional changes to the model.

<sup>4</sup><https://www.theguardian.com/us-news/2016/feb/04/san-francisco-then-and-now-super-bowl-50>

<sup>5</sup><http://keras.io>

<sup>6</sup><http://github.com/healpy/healpy>

<sup>7</sup><http://healpix.sourceforge.io>

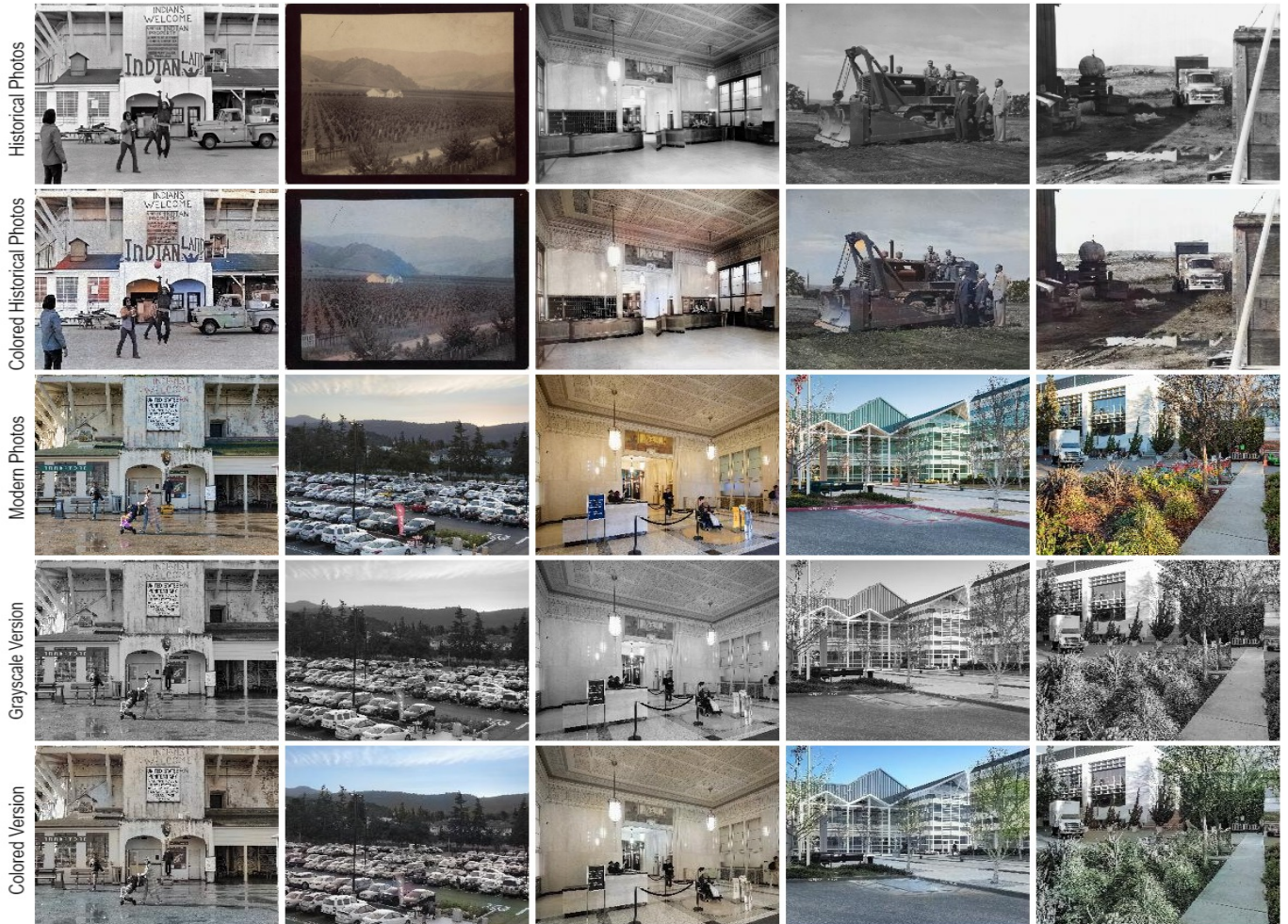


Figure 4: Examples of modern versus historical photos of San Francisco.

	New York	San Francisco
Number of Flickr photos	1139379	865977
Number of historical photos	37618	13255
Area of bounding box( km <sup>2</sup> )	2303.27	2525.01
Number of HEALPix cells (Flickr)	820	421
Number of HEALPix cells (Historical)	328	63

Table 2: Statistical characterization of the evaluation datasets.

## 4 Experimental Evaluation

This section details the experimental evaluations conducted on the networks introduced in Section 3. Specifically, Section 4.1 describes the datasets and the evaluation methodology used in the experiments. Section 4.2 presents the results of evaluating the different types of architectures, trained to perform the geocoding task, using datasets with photos from both cities of San Francisco and New York.

### 4.1 Datasets and Evaluation Methodology

The results reported in Table 3 and Table 4 were obtained by testing the models over 3 datasets, namely the Yahoo Flickr Creative Commons 100 Million dataset, the Old San Francisco Project dataset and the Old New York Project dataset. Instead of using the entire Flickr dataset, two bounding boxes were defined, corresponding to the cities of New York and San Francisco, which originated two subsets, one from each city. Additional information about the three datasets and the bounding boxes is described in Table 2. Figure 6 illustrates heatmaps concerning the distribution of the photos in the datasets, over the cities of New York and San Francisco.

### 4.2 Results

Table 3 presents the results obtained with photos from the area of San Francisco, while Table 4 presents the results for the area of New York. In both cases, the tables report on the percentage of photos for which the estimated coordinates

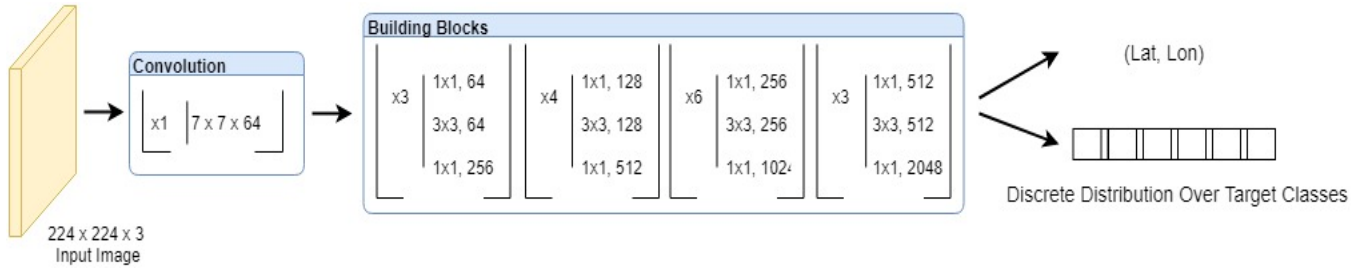


Figure 5: Illustration of the geocoding network combining classification and regression outputs.

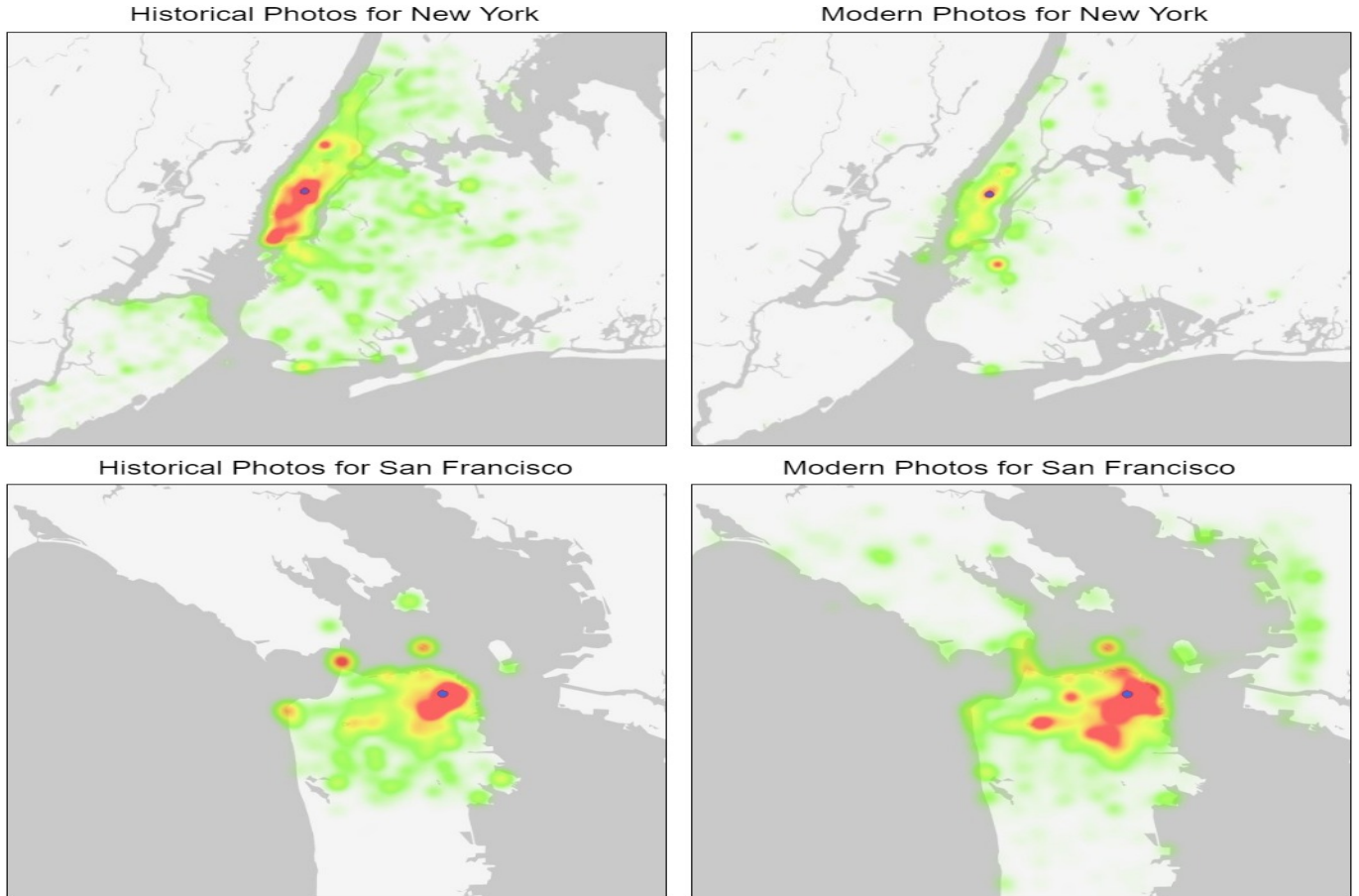


Figure 6: Density of photos in each of the evaluation datasets, together with the centroid for the HEALPix cell with the highest density.

are within a given distance threshold of the ground-truth coordinates, and also the mean and median distances between the estimated coordinates and the ground-truth. The distances were measured using the Vincenty’s geodetic formulae, and the tables compare the proposed method against both following sets: the baseline approaches, and the model variations corresponding to ablation tests.

Regarding the baselines, I report the results: 1) for a random assignment of coordinates within a bounding box covering the coordinates of both historical and modern photos in each dataset; and 2) for the assignment of all pictures to the centroid coordinates of the HEALPix cell with more photos in each dataset. The partitioning resolution applied in the Earth’s division to get the cells was the same as the one used in the following reported experiments. This document also presents the results obtained by the proposed method with modern photos collected from Flickr, specifically comparing models based on a regression loss, a classification loss, or a combination of both.

In terms of the ablation tests, I report on models trained directly on the original historical photos, or trained on historical photos but extending the neural network architecture with layers that attempt to color the historical photos, prior to feeding the information to the part of the network that does the geocoding. It was also tested the accuracy of models trained on modern Flickr data for assigning coordinates to the historical photos, either using the historical photos directly or attempting to colorize them first. Finally, the complete end-to-end model was pre-trained with modern Flickr data and then refined with historical photos, also integrating neural network layers for coloring the historical images.

Concerning the results of the experiments reported in Table 3, and starting with the geocoding networks trained directly on modern and historical data (2<sup>nd</sup> and 4<sup>th</sup> set in Table 3), the regression networks outperformed the other two

	Accuracy				Distance (Km)	
	<100m	<500m	<1Km	<5Km	Mean	Median
Flickr Random	0.002	0.025	0.117	3.043	31.584	28.803
Flickr Most Frequent Cell (MFC)	0.156	<b>3.558</b>	12.587	71.557	<b>4.342</b>	3.212
Flickr MFC (fine-grained partition)	<b>1.077</b>	8.003	<b>16.756</b>	70.125	4.456	3.192
Flickr MFC (coarse-grained partition)	0.005	0.646	4.237	<b>77.070</b>	4.357	<b>3.057</b>
Flickr Regression (R)	0.063	1.593	6.185	69.714	4.711	3.531
Flickr Classification (C)	0.134	2.221	8.219	67.183	4.895	3.590
Flickr Combined (R/C)	0.065	1.517	5.900	66.947	4.929	3.696
Hist. Random	0	0.045	0.151	2.958	31.204	28.263
Hist. MFC	0.913	6.511	<b>24.738</b>	74.772	3.656	2.218
Hist. MFC (fine-grained partition)	<b>0.943</b>	<b>11.181</b>	22.641	75.141	3.712	2.248
Hist. MFC (coarse-grained partition)	0.030	0.958	5.915	78.250	3.600	2.771
Historical R	0.098	1.788	6.878	<b>78.514</b>	3.829	2.941
Historical C	0.460	5.334	15.987	68.487	4.307	3.101
Historical R/C	0.166	3.282	10.977	73.030	4.284	2.861
Coloring + Hist. R	0.038	1.690	7.349	78.582	3.772	2.853
Coloring + Hist. C	0.325	4.142	13.052	65.613	4.595	3.519
Coloring + Hist. R/C	0.075	2.271	8.563	77.496	3.856	2.779
Flickr + Historical R	0.023	2.309	8.050	74.840	4.027	3.136
Flickr + Historical C	0.309	3.508	11.279	71.415	4.170	3.087
Flickr + Historical R/C	0.098	1.788	7.016	71.792	4.237	3.383
Flickr + Col. + Hist. R/C	0.189	6.805	18.657	73.776	<b>3.536</b>	<b>2.114</b>
Flickr Pre-Training	0.068	2.165	8.389	70.796	4.162	3.370

Table 3: Experimental results with photos from San Francisco.

	Accuracy				Distance (Km)	
	<100m	<500m	<1Km	<5Km	Mean	Median
Flickr Random	0.004	0.037	0.139	3.396	19.789	19.739
Flickr Most Frequent Cell (MFC)	0.050	5.697	<b>12.836</b>	53.393	<b>6.075</b>	<b>4.493</b>
Flickr MFC (fine-grained partition)	0.007	<b>6.332</b>	7.072	30.517	8.165	7.645
Flickr MFC (coarse-grained partition)	0	0.428	0.949	44.619	7.118	5.608
Flickr Regression (R)	<b>0.078</b>	1.764	6.250	<b>53.786</b>	6.255	4.613
Flickr Classification (C)	0.030	1.967	5.971	44.582	7.288	5.592
Flickr Combined (R/C)	0.069	1.595	5.689	52.489	6.395	4.748
Hist. Random	0.003	0.037	0.149	3.390	20.217	19.955
Hist. MFC	0	<b>3.825</b>	<b>10.460</b>	45.449	<b>7.161</b>	5.493
Hist. MFC (fine-grained partition)	<b>0.080</b>	2.012	3.015	22.345	9.852	8.476
Hist. MFC (coarse-grained partition)	0	0	0.872	38.397	8.269	6.287
Historical R	0.064	1.042	3.857	46.515	7.274	5.391
Historical C	0.024	1.327	4.418	34.130	9.082	7.536
Historical R/C	0.046	1.133	4.447	46.425	7.355	5.475
Coloring + Hist. R	0.016	0.588	2.430	42.161	7.585	5.817
Coloring + Hist. C	0.013	0.909	2.930	29.736	10.023	8.559
Coloring + Hist. R/C	0.032	0.753	2.773	43.612	7.535	5.719
Flickr + Historical R	0.066	1.706	5.760	45.192	7.486	5.693
Flickr + Historical C	0.035	1.446	4.732	38.979	8.209	6.542
Flickr + Historical R/C	0.077	1.536	5.561	44.649	7.503	5.785
Flickr + Col. + Hist. R/C	0.016	0.718	2.999	<b>47.347</b>	7.451	5.409
Flickr Pre-Training	0.006	0.423	2.199	47.158	7.306	<b>5.340</b>

Table 4: Experimental results with photos from New York.

types of architectures. It was expected that the networks trained to output both classification and regression values would performed the best, since the weights were being updated using two loss functions. In the classification networks, the evaluation instances of both modern and old historical sets were entirely assigned to the HEALPix cell containing most photos, and to the surrounding cells. On the other hand, most of the inferred coordinates by the regression

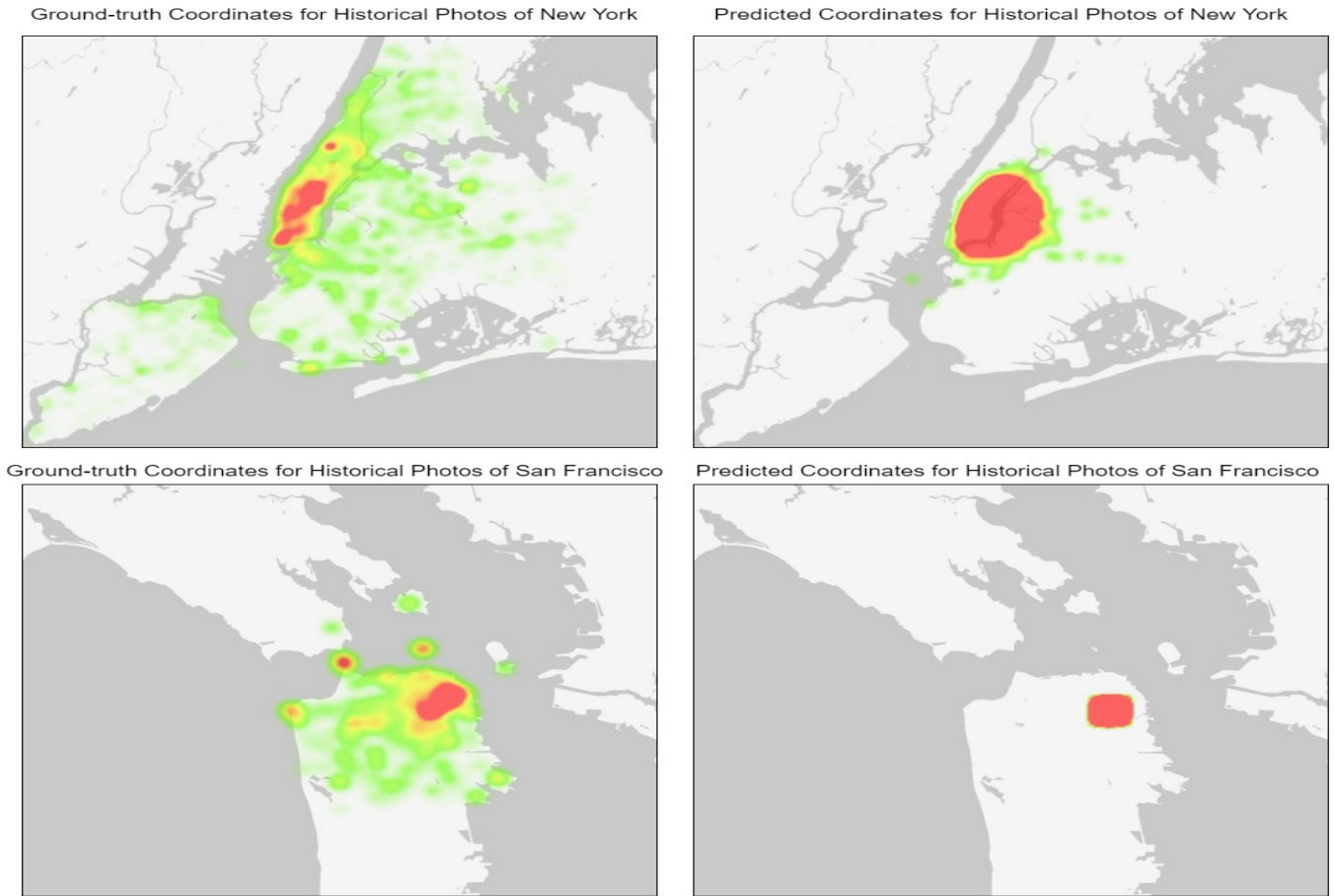


Figure 7: Density maps with basis on the ground-truth coordinates of the historical photos, versus the estimated coordinates by a neural network combining regression and classification losses.

networks covers the coastline, which contains a big concentration of photos, and may have contributed to the success of the results. However, in the regression + classification (combined) networks, the result’s distribution moved from the coastline towards an area corresponding to the same HEALPix cell containing the majority of the photos. From this analysis, one can conclude that having a big number of photos in a single HEALPix cell is impacting the results, specifically the networks trained with both classification and regression + classification losses. Attend to Figure 6 to see the location of the most frequent cell, marked with a blue dot.

Evaluating historical photos directly on networks trained with modern data proved to be not as effective as using a colored component to first colorize those historical photos, as highlighted when comparing the results of the *Flickr + Historical* experiments, with the ones from *Flickr+Col+Hist R/C* and *Flickr Pre-Training* end-to-end networks. Regarding these two last experiments, results indicate that the end-to-end network not re-trained with historical data performs better than the same end-to-end network re-trained with historical data. When analyzing the heatmaps with the inferred coordinates distribution, illustrated in Figure 7, it is shown that the predicted coordinates of the *Flickr+Col+Hist R/C* network are all gathered in a single area, which corresponds, once again, to a location near the HEALPix cell containing the most photos. Additionally, Figure 8 shows that the error increases as the photos get far from the area where the HEALPix cell containing more photos is located. Therefore, one can conclude that having a significant percentage of pictures in a single HEALPix cell affects the accuracy of the models, as they only learn to geocode photos in a single area.

Regarding the city of New York, from training and evaluating the old and modern photos directly on the networks, without coloring them, it was concluded that once again, the regression network performed better in both datasets, in comparison to other tested network typologies. Resembling the experiments on the San Francisco’s datasets, the classification networks predicted coordinates in the HEALPix cell containing the most photos, and in the surrounding cells. The combined network trained on the Flickr dataset predicted coordinates covering a much larger area, in comparison to the regression network, which may have also led to higher error distances, and thus making the results from the regression network the best ones.

Once again, it was proved that evaluating historical photos in networks trained with modern data produces better results when those old photos are first colorized. With that said, it is clear the importance of the colorizer component in the success of the task. Regarding the end-to-end network, experiments concluded that the network re-trained with historical data performs the best, contrary to what happens in the experiments on the end-to-end networks with San Francisco’s datasets. The heatmap of the corresponding results distribution is illustrated in Figure 7, and contrary to the best performing model on photos from San Francisco, the results are more distributed, covering both overland and

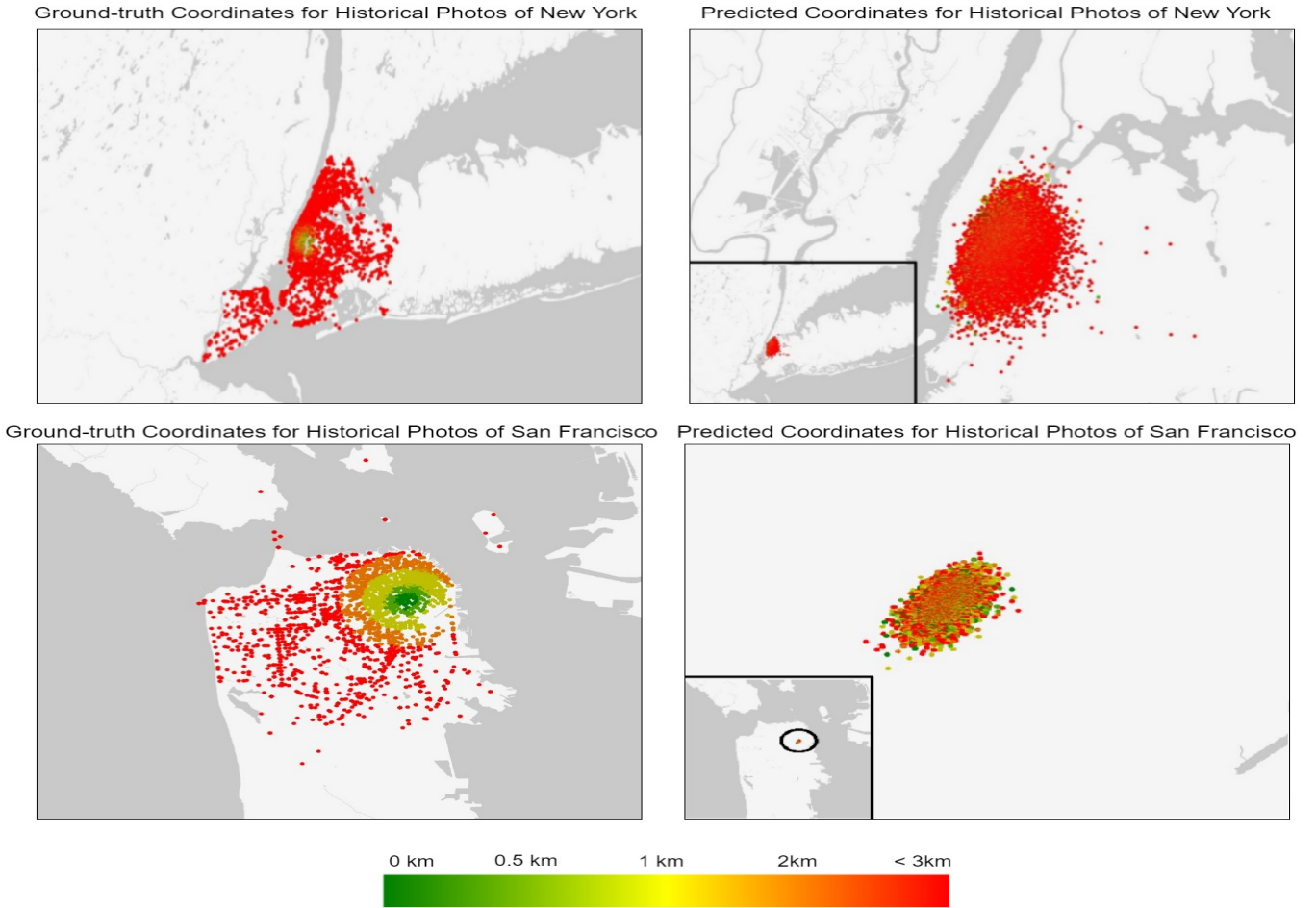


Figure 8: Error distribution with basis on the ground-truth coordinates of the historical photos, and the estimated coordinates by a neural network combining regression and classification losses.

maritime regions, and are not uniquely concentrated around the HEALPix’s most frequent cell. However, the error distribution from Figure 8 shows that the error is lower in the area of the HEALPix’s cell containing the most photos, and increases significantly in the remaining areas of the city. Once again, the impact of having a big number of photos in a single region reflects on the achieved results.

None of the trained and evaluated end-to-end networks for both cities were capable of significantly outperforming the *most frequent cell* baseline. This is a solid baseline, as there is a significant number of instances in a single cell. For instance, when changing the resolution of the partitioning in order to get HEALPix cells with smaller dimensions, the baseline error values increased significantly in both datasets, as it is reported in both Table 3 and 4. When using a coarse-grained partition in order to get HEALPix cells with bigger dimensions, the error values also increase in comparison with the baselines using the same partitioning resolution as in the experiments, but not as much as when using the fine-grained partitioning. The network not fine-tuned with historical data from San Francisco achieves slightly better results than the baseline, however, the difference between the mean values is less than 100 meters.

## 5 Conclusions and Future Work

This work presented an approach fully-based on CNNs to address the task of automatically geocoding historical photos for the cities of New York and San Francisco, introducing a novel end-to-end network aiming for this task. The network consists of having a first CNN trained to make color-wise image transformations over grayscale images, connected to a second CNN trained to geocode photos. This is a challenging task, which reflects on the results, as the end-to-end networks were not able to outperform the baselines. Some of these baselines contain really low distance error values, the image quality of the historical photos is significantly low, and the site’s landscape changes over time, making geocoding historical photos on the current scenery a daunting task. It was proved that the introduced end-to-end network is more efficient for the task in comparison to other networks where there is no coloring component and the query photos are directly processed by the geocoding layers.

As future work, it would be interesting to replace the colorizer network with a more complex CNN, such as the one introduced by Zhange et al. [15], in which a grayscale input image is mapped to a distribution of color values. There is yet another approach that would be an alternative to the current colorizer component, introduced by He et al. [16], in which a grayscale photo is colorized based in a referential photo. Concerning the geocoding component, Walch et al. [17] proved that using LSTM units to perform dimensionality reductions on the outputs of feature vectors lead to a

better localization performance. Thus, it would be interesting to add those units to the current network’s architectures, and train them with the same training instances used in this work.

**Acknowledgements** This research was supported by Fundação para a Ciência e Tecnologia (FCT), through the project grant with reference CMUPERI/TIC/0046/2014 (GoLocal), as well as through the INESC-ID multi-annual funding from the PIDDAC programme, which has the reference UID/CEC/50021/2013. I also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used in the experiments.

## References

- [1] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2), 2016.
- [2] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-propagating Errors. *Cognitive Modeling*, 5(1), 1988.
- [3] James Hays and Alexei A. Efros. IM2GPS: Estimating Geographic Information from a Single Image. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
- [4] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-Photo Geolocation with Convolutional Neural Networks. In *European Conference on Computer Vision*, 2016.
- [5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.
- [6] Nam Vo, Nathan Jacobs, and James Hays. Revisiting IM2GPS in the Deep Learning Era. In *Computer Vision, 2017 IEEE International Conference on*, 2017.
- [7] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, (1), 2014.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceeding of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.
- [9] Lucas Rodes-Guirao Federico Baldassarre, Diego Gonzalez-Morin. Deep-Koalarization: Image Colorization using CNNs and Inception-ResNet-v2. *ArXiv:1712.03400*, (1), 2017.
- [10] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2017.
- [11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*, abs/1704.04861(1), 2017.
- [12] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [13] Thaddeus Vincenty. Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations. *Survey Review*, 23(1), 1975.
- [14] Krzysztof M Gorski, J E Felten, Benjamin D Wandelt, Frode K Hansen, Eric Hivon, and Anthony J Banday. The HEALPix Primer. *arXiv.org*, astro-ph(2), 1999.
- [15] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful Image Colorization. In *Proceedings of the European Conference on Computer Vision*, 2016.
- [16] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep Exemplar-based Colorization. *ACM Transactions on Graphics*, 37(4), 2018.
- [17] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based Localization using LSTMs for Structured Feature Correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.