Detecting languages in streetscapes using deep convolutional neural networks

You Xuan Thung, Tom Benson, Nikita Klimenko
Senseable City Lab
Massachusetts Institute of Technology
Cambridge, MA, USA
Email: thungyx@mit.edu, bensont@mit.edu, klimenko@mit.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020745

***Abstract*—Cities are laden with visual clues—public road signs, advertising billboards, street names, place names, street art, and commercial shop signs. In urban studies literature, it is common to use such clues, littered along streetscapes, to understand cities better. Sociologists have tried to derive meaning from the linguistic diversity in streetscapes, but their efforts are limited by the laborious effort of field studies. Taking advantage of the availability of street view imagery (SVI) today and the advent of deep learning driven by the availability of big data, we undertake a machine learning approach to quantify the linguistic diversity in cities. In this paper, we introduce a language detection tool to identify the presence of English, Swedish, Arabic and Chinese in streetscapes. The tool is built on a pretrained DenseNet-121 model and trained with both synthetic images, and real streetscapes scraped from Google Street View (GSV). We achieve a test accuracy of 80.8% across all four languages, surpassing the performance of best performing optical character recognition (OCR) tools by 9 percentage points. The F1 score of 79.8% also surpasses that of present OCR tools by 11 percentage points, which suggests that our model is both accurate and reliable. We use the gradient-weighted class activation map (Grad-CAM) to show that our model is visually interpretable. We then apply the tool to Stockholm, Sweden to create a portrait of the linguistic diversity of the city and find a chasm between linguistic diversity and population diversity, which points to the importance of measuring the two measures separately.**

***Index Terms*—deep learning; computer vision; language detection; convolutional neural networks**

I. Introduction

Any visitor to New York City would hardly find it difficult to locate an ethnic enclave. Chinatown, Little Italy, Koreatown, Little Egypt are all easily identifiable on a map, and anyone who unknowingly walks into these places would quickly figure out where they are from street signs, names of restaurants, advertisements and other language clues (Figure 1). Yet, not all enclaves have self-explanatory names like those in a major city. Nothing about the names Skärholmen (Stockholm) or Edgware Road (London) suggests that they have a large Arabic-speaking population, but visual clues from the languages that appear on these streets suggest otherwise.

Although census data may provide fine-grained information on the ethnolinguistic distribution of residents across census tracts, visual information provides a different perspective, giving us clues not only about the people who live there but also about people who work in and frequent the area. When combined with census information, visual clues may give us an idea of the level of social integration within an area, and whether linguistic diversity is commensurate with the ethnic diversity of the urban area. From an urban policy standpoint, it is important to be cognizant of the ethnolinguistic make-up and linguistic diversity of urban regions and monitor how these metrics evolve spatiotemporally. Therefore, we seek to use modern computer vision tools to quantify the linguistic diversity of streetscapes.

Fig. 1. Streetscape of Chinatown, Little Italy, Little Egypt, Koreatown from top left in clockwise direction.
Sources: "Chinatown, NYC" by nmadhu2k3, "Little Italy, NYC" by Robert-Francis and "Koreatown NYC" by Chun's Pictures are licensed under CC BY 2.0. Image of Little Egypt is taken from GSV.

This study is especially interesting in Stockholm and Sweden more generally. Sweden has been characterized by a common language, religion, and political history for most of its history [1]. In the early 2010s, large waves of immigration occurred in Stockholm, including the influx of Arabic-speaking people in the aftermath of the Arab Spring. This influx saw increases in social segregation and social unrest, where areas of ethnic segregation have become more apparent. These minorities have entered Stockholm and other Scandinavian countries but have not necessarily felt comfortable expressing their culture through language, local stores, or signage [2]. With this in mind, it is interesting to see how native Swedes

The authors would like to acknowledge the financial support from the Senseable Stockholm Lab funded by the City of Stockholm, KTH Royal Institute of Technology, the Stockholm Chamber of Commerce and Newsec, in collaboration with MIT Senseable City Lab.

cope with living with a growing population of people who look and speak differently and how this manifests in the linguistic diversity on the streets.

Current tools that can be used for language classification are generally developed for optical character recognition (OCR), and the performance is mixed. Therefore, we take a different approach—rather than trying to identify every character in a scene, we seek to detect the presence of specific languages. Given that we are only interested in the prevalence of languages in a city rather than the actual words used, we can sacrifice the complexity of the problem to achieve better accuracy.

Since we aim to study linguistic diversity in streetscapes in Stockholm, we construct a novel multilingual dataset containing Google Street View (GSV) images from seven cities, covering the presence of four languages—English, Swedish, Arabic and Chinese. We train a binary classification network over four independent tasks using a pre-trained DenseNet, and our best model achieves a test accuracy of 80.8% with a corresponding F1 score of 79.8%.

We apply this model to GSV images in Stockholm to further test the model's usefulness. We find that the spatial distribution of language concentration in Stockholm agrees with intuition. Although somewhat counterintuitive, we find that linguistic mix does not correspond well with population mix,[1] instead providing a different perspective of Stockholm. We argue that in a culturally homogeneous society such as Sweden, minorities may not necessarily feel comfortable expressing themselves in their languages, which is reflected in the chasm between population and linguistic mix.

In summary, we create a novel dataset, introduce a different paradigm for detecting languages in streetscapes, and show that quantifying linguistic diversity can provide another perspective of cities that existing metrics do not.

II. Literature Review

A. Linguistic Diversity of Streetscapes

The study of linguistic diversity in streetscapes is more commonly known as the study of linguistic landscapes in sociolinguistic literature. This body of literature is motivated by the abundance of linguistic features littered across streetscapes—public road signs, advertising billboards, street names, place names, commercial shop signs, street art, and public signs on government buildings [3]. As Gorter eloquently puts it: "[s]igns are everywhere, they permeate our daily life, and they can give us a sense of place" [4]. However, as much as signs influence us and our behaviors, they are also artifacts of individual and social preferences [5]. After all, the languages used on a sign is an outcome of deliberate choices made by political or economic actors, which are in turn influenced by both economic interests and the social preferences of the populace. In a case study of Donostia, Spain and Ljouwert, the Netherlands, Onofri *et al.* find that the choice of languages in a sign is strongly influenced by the type of establishment associated with the sign. For instance, the use of English is strongly positively correlated with an establishment being an international chain. In contrast, the use of Frisian in Ljouwert is strongly correlated with the establishment being a shop or an official building [5].

As much as we can learn something about individual and social preferences from the languages we see in street signs, we may also learn something from those we do *not* see. In a field study of Oslo, Norway, Opsahl finds scarcely any presence of the Polish language, despite the large number of Polish immigrants in Norway [6]. Although we often see cities as platforms for languages to "manifest their vitality as well as their visibility" [7], the counterintuitive "invisibility" of Polish makes one question if individuals can indeed express themselves in their preferred tongue in Oslo [6]. And perhaps this is a valid question not just in Norway but also in Sweden, with Leinonen and Toivanen arguing that the Nordic nations have an identity that builds on cultural, religious and linguistic homogeneity [8].

Even though studying linguistic landscapes is interesting, research in this area often relies on extensive field work in small communities. Such an approach offers deep insights into specific urban areas, but these insights may not necessarily generalize to the larger urban environment. Given the abundance of street view imagery (SVI) offered by GSV and similar services, there is a nascent field that seeks to automate the process of characterizing linguistic landscapes. In particular, Hong conducted a proof-of-concept study of a small Chinese community in Seoul, South Korea using Google Vision API tools [9]. Albeit their approach sets a new direction for studying linguistic landscapes, they find that many word sequences in their samples were unrecognized by the algorithm. To advance this literature further, we also take a big-data approach in constructing fine-grained maps of linguistic diversity in Stockholm.

B. Scene Text Recognition

The problem of scene text recognition is part of a larger field of image-based sequence recognition. The fundamental motivation in this field of research is to extract as much high-quality information from images as possible. Therefore, current tools that can be used for language classification are designed to not only detect if a language is present in an image but also output what the specific word sequences are.

EasyOCR[2] is a popular open source ready-to-use tool for scene text recognition. The tool uses a convolutional recurrent neural network (CRNN) architecture [10] that integrates feature extraction, sequence modeling and transcription into a unified framework to identify text in a scene. EasyOCR supports over 80 languages and has reasonable performance for English. However, it has a distinctly poorer performance for

[1]Since fine-grained data on ethnic/racial distribution is not publicly available, we construct an entropy measure with broad categories used in Statistics Sweden's publicly available data (Section VI-B).

[2]Source code available at: https://github.com/JaidedAI/EasyOCR

identifying other languages (see Section V for a comparison with our model).

A host of commercial OCR tools also spells more promise, although it is unclear how these tools function. We experiment with Google OCR, which does a reasonably good job of recognizing specific characters when it detects the presence of any word sequence. However, it has a low recall, which corroborates with the findings of Hong [9]. This translates to an average accuracy and an F1 score poorer than EasyOCR (see Section V). The mixed performance of state-of-the-art OCR tools motivates adopting a different paradigm in scene text recognition.

C. Classification with CNNs

Therefore, we seek to tackle an easier task—identifying the presence of a language in a scene. Like how OCR tools use some form of convolutional neural network (CNN) to extract features, our language detection model is also built on a pre-trained CNN.

Common CNNs used in extracting features from a scene include VGG [11] and ResNet [12]. While VGG offers a more parsimonious network (19 layers), the increased complexity of ResNet (>100 layers) accords more degrees of freedom for better feature representation. Against the backdrop of deeper and deeper networks and the associated problem of vanishing gradients, the Dense Convolutional Network (DenseNet) [13] offers a different modeling paradigm. In each layer of a DenseNet, the feature maps of all preceding layers are used as inputs, thereby creating a *denser* representation structure with fewer layers, striking a balance between model parsimony and representation space. DenseNets have performed well in image and scene classification tasks [14], [15], and we use a pre-trained DenseNet in our implementation.

III. Data

A. Synthetic Data

Given the labor cost of manual labeling, we automate the process of data generation by using SynthText,[3] a tool for generating text onto given background images. Generating synthetic data is a common technique used in scene text recognition [16], [17] due to limited authentic data. To synthesize images with text, Gupta *et al.* identify regions with sufficient continuities using segmentation data and transform the text to be placed on images using depth data [18].

Although SynthText was originally developed only for English, we made amendments to the original code to provide support for generating Swedish, Arabic and Chinese text. We do this by introducing a multilingual corpus and fonts that support the three other languages.

Given that we can change the language of the text generated while controlling for the background image, the data generated may skew training towards focusing on the text rather than irrelevant visual features. In Figure 2, we provide examples of four images with exactly the same background, each with text of a different language. We provide statistics of the synthetic dataset in Table I.

Fig. 2. Examples of synthetic images generated with the same background but with different languages—Arabic, Chinese, Swedish and English, from top left in clockwise order.

B. Google Street View Images

Although there are benefits in using synthetic data, we recognize that visual domain adaptation is difficult [19]. Since we ultimately want to apply our text detection tool to real scenes, it is important for the model also to be trained on real streetscapes. Therefore, we source real data from GSV and manually label them with `pigeon`,[4] an open source labeling tool. With the intention of applying this tool to Stockholm, we scrape images from densely populated areas with a good amount of text in the scene. Our dataset comprises images from cities where the target languages are dominant—Stockholm for Swedish, Ramallah, Bethlehem and Beirut for Arabic and Hong Kong for Chinese. To avoid overfitting on these cities where the target languages are highly prevalent, we also include images from London and New York City, both global metropolis where minority languages feature in a less prominent manner. Therefore, we have a dataset with good coverage of the four target languages, with variations in how prominently they are featured and the architectural styles of the scenes they are featured in. This ensures that the resultant model is generalizable to our case study of Stockholm.

To obtain GSV images, we begin by generating sampling points along the road network in areas of interest at 50-meter intervals using OpenStreetMap. We then make API requests from GSV using these sampled coordinates and the following parameters—90° field-of-vision, 0° pitch, 50m radius. For each set of coordinates, we obtain images at compass headings of 0°, 90°, 180° and 270°, thereby capturing the full panorama at each point. The summary statistics of data obtained from each city is presented in Table II.

We first approached the study with the hypothesis that pre-training with synthetic data would allow for better overall performance. Therefore, we generated synthetic data and labeled

[3]Source code available at: https://github.com/ankush-me/SynthText

[4]Source code available at: https://github.com/agermanidis/pigeon

TABLE I
SIZE OF DATASET FOR EACH LANGUAGE

Language	Synthetic	Real
English	1028	6105
Swedish	1028	1175
Arabic	1028	1434
Chinese	1028	1165
None	1028	6049
Total	5140	15928

Notes: The synthetic data for each of the four languages is generated from the same 1028 background images. After finding good performance with training solely on real data, we scaled up manual labeling of real data, thereby leading to a much larger dataset of real data.

TABLE II
SIZE OF DATASET BY CITY

City	En	Sv	Ar	Cn	None	Total
London	2836	0	103	0	2543	5383
New York City	355	0	29	0	302	660
Stockholm	590	1175	2	15	1195	2633
Ramallah	873	0	1048	0	988	2214
Bethlehem	188	0	217	0	240	522
Beirut	48	0	35	0	59	127
Hong Kong	1215	0	0	1150	722	2034
Total	6105	1175	1434	1165	6049	13573

a small amount of real data manually. After finding better performance with training solely on real data vis-à-vis pre-training with synthetic data, we scaled up manual labeling of real data, thereby leading to a much larger dataset of real data (Table I).

C. Dealing with Imbalanced Data

Since the language detection model is construed as a binary classification model with four parallel tasks, we will have many more negative examples than positive examples. For example, if we train the classifier solely on synthetic data where each image only has one language present, each classifier will have 1028 positive examples and 4112 negative examples. Therefore, we downsample the data randomly before training so that the number of positive examples and negative examples are equivalent.

IV. BUILDING A COMPUTATIONAL MODEL

A. Hyperparameters

We train the models using an NVIDIA GeForce RTX 2080 Ti GPU with 11GB memory. We adopt a train-validation-test split of 70-15-15 and experiment with different learning rates before using the model with the best validation accuracy on our test set. The remaining hyperparameters are listed in Table III.

The base code for training and testing follows a modified version of the PyTorch implementation from [20] that supports multi-label classification.[5]

[5]Modifications made by Fan Zhang from MIT Senseable City Lab.

TABLE III
HYPERPARAMETERS

Hyperparameter	Choice
Base Architecture	DenseNet-121
Learning Rate	$0.0005, 0.001, 0.002$
Momentum	0.9
Weight Decay	10^{-4}
Batch Size	100
Epochs	150
Loss Function	Cross-entropy
Optimizer	Stochastic gradient descent
Image size	256×256

B. Training Process

We first train the model purely on the synthetic dataset, before training the best performing model on the real dataset for another 150 epochs. We also train the model with the real dataset from scratch.[6]

C. Second-order Metrics

The model predicts if a language is present in an image (outputs 1) or not (outputs 0), lending itself to a measure of language concentration in an urban area as defined as:

$$P_{x,\ell} = \frac{1}{|x|} \sum_{i \in \mathcal{S}(x)} \mathbb{1}(\text{language } \ell \text{ is in image } i) \tag{1}$$

where $\mathcal{S}(x)$ is the set of images in an urban area x. To quantify the linguistic diversity in an urban area, we use the concept of Shannon entropy from statistical physics that is commonly applied to capture the notion of diversity [21]–[23].

$$E_x = \sum_{\ell} -P_{x,\ell} \log(P_{x,\ell}) \tag{2}$$

D. Evaluation Metrics

To evaluate the performance of our model, we look at both classification accuracy and the F1 score, which is defined as such:

$$\text{Precision}_\ell = \frac{TP_\ell}{TP_\ell + FP_\ell} \tag{3}$$

$$\text{Recall}_\ell = \frac{TP_\ell}{TP_\ell + FN_\ell} \tag{4}$$

$$\text{F1}_\ell = 2 \cdot \frac{\text{Precision}_\ell \cdot \text{Recall}_\ell}{\text{Precision}_\ell + \text{Recall}_\ell} \tag{5}$$

where TP_ℓ (FP_ℓ) is the number of positive examples that are (in)correctly classified for language ℓ while FN_ℓ is the number of negative examples that are incorrectly classified for language ℓ. Although classification accuracy is easily interpretable, it is not robust to data imbalance. Since we have a larger proportion of negative examples than positive examples for our test dataset, we are concerned about misleading test statistics (for example, a model may achieve high test accuracy by classifying everything as negative). Tracking recall and the F1 score provides us with another perspective of the model's performance.

[6]i.e. from pre-trained DenseNet-121 parameters

V. Results

A. OCR vs Our Method

The test accuracy of the different models is presented in Table IV. We use our own test dataset, comprising 2964 images scraped from GSV, to evaluate the performance of our model and compare them to those of ready-to-use OCR tools. We find that both OCR tools have around 60-70% accuracy across the four languages.

From the precision and recall of the different models (Tables V and VI), we find that the OCR tools are generally quite conservative, as precision is generally much higher than recall across the four languages. This is especially the case for Google OCR, which has almost no false positives, at the expense of a large number of false negatives.

In contrast, we find that our language classifier works well for all four languages, when trained with real data. The overall accuracy of both models trained with real data is higher than that of both OCR tools and crucially, the gap between precision and recall for our models is small. In fact, for the models trained with real data, the gap between precision and recall for each classifier only has a maximum of 11.5 percentage points, much smaller than those of the two OCR tools. Consequently, the F1 scores of the models trained with real data are also higher than those of the OCR tools (Table VII).

Intuitively, the superior performance of our model in comparison with OCR tools arises from the fact that it is not trained to care about the specific characters. In an OCR tool, the model is trained to identify word sequences and only yields any output if the model has a sufficiently high confidence that a particular word sequence is present in the image. The language identified in the word sequence is only a byproduct and depends on a word sequence being identified first. However, identifying a language does not necessarily require identifying the specific words first, even for humans, particularly when the languages of concern are distinct from one another. In our specific use case, we note that Arabic is a cursive language where most of the characters in one word are connected, while written Chinese comprises pictographs that are of roughly equal size. These characteristics make Arabic and Chinese distinct from the Latin script and from each other. Even between English and Swedish, there are clear visual distinctions, with the use of accents and much higher frequency of long compound words in Swedish (e.g. Centralstationen (Swedish) vs Central Station (English)). Not focusing on specific characters allows the model more degrees of freedom to focus on distinct linguistic features, and to output a positive result, even when it is not clear what the specific words are (e.g. in scenes where the words are small or skewed).

B. Training with Synthetic vs Real Data

The performance of the model trained solely on the synthetic data is poor, with the test accuracy being about as good as a random guess. This points to the limitation of transfer learning—the underlying distributions of the synthetic data and real data are different and a model trained solely on synthetic data may not generalize well to real data.

However, once we train the model further with real data, we find improved performance and the test accuracy for languages other than English is much higher than that of the OCR tools. We also trained the model from scratch with real data and this model achieves better performance than OCR tools across all four languages, with a total test accuracy of 80.8%. Although we hoped that pre-training with synthetic data would teach the model to focus on the text rather than irrelevant features and therefore lead to better performance than training from scratch, it is likely that the poor generalization of the synthetic data made the parameters learned from training with synthetic data a distraction rather than an aid. The strong performance of training from scratch suggests that it may not be necessary to use synthetic data to train a language detection model.

C. Visual Interpretability

Although our model performs well, we want to be sure that it is looking at the right features instead of picking up spurious correlations. Therefore, we use a gradient-weighted class ac-

TABLE IV
Test accuracy of EasyOCR, Google OCR and our models

Model	En	Sv	Ar	Cn	Total
EasyOCR	75.2	67.0	68.6	64.0	71.9
Google OCR	71.6	67.6	74.7	62.6	70.5
Synthetic	55.3	53.4	48.1	53.1	53.9
Synth + Real	71.6	82.4	**88.6**	88.9	77.2
Real	**76.4**	**85.5**	88.1	**91.1**	**80.8**

Notes: For each training paradigm, we only include the test accuracy of the model with the highest total validation accuracy. Highest test accuracy bolded.

TABLE V
Precision of EasyOCR, Google OCR and our models

Model	En	Sv	Ar	Cn
EasyOCR	82.2	65.7	68.7	66.1
Google OCR	71.2	94.1	100.0	98.0
Synthetic	68.7	42.9	47.2	60.3
Synth + Real	69.2	76.2	86.7	87.6
Real	79.2	79.0	85.6	90.2

Notes: For each training paradigm, we only include the precision of the model with the highest total validation accuracy.

TABLE VI
Recall of EasyOCR, Google OCR and our models

Model	En	Sv	Ar	Cn
EasyOCR	63.3	56.6	65.1	60.9
Google OCR	71.2	30.2	47.8	27.4
Synthetic	18.9	9.4	57.4	24.6
Synth + Real	76.4	88.7	90.4	91.1
Real	70.7	92.5	90.9	92.7

Notes: For each training paradigm, we only include the recall of the model with the highest total validation accuracy.

TABLE VII
F1 SCORE OF EASYOCR, GOOGLE OCR AND OUR MODELS

Model	En	Sv	Ar	Cn	Total
EasyOCR	71.5	60.8	66.9	63.4	68.4
Google OCR	71.2	45.7	64.7	42.8	63.4
Synthetic	29.6	15.4	51.8	34.9	30.8
Synth + Real	72.6	82.0	**88.5**	89.3	77.8
Real	**74.7**	**85.2**	88.2	**91.4**	**79.8**

Notes: For each training paradigm, we only include the F1 score of the model with the highest total validation accuracy. Highest F1 score bolded.

tivation map (Grad-CAM) to translate gradient information of each of the four classifiers flowing into the final convolutional layer onto a heatmap.[7] In Figure 3, we present the heatmaps produced for a sample of correctly classified images in the test dataset. In general, we find that in correctly classified examples, the model focuses on the right things—store signs, road markings. In incorrectly classified examples (Figure 4), the model might still be looking at the right things but fails to detect a language likely because the word sequences are too small or indistinct.

We also provide the Grad-CAM of correctly classified multilingual scenes (Figure 5 provides five examples—English and Swedish, English and Arabic, English and Chinese, Swedish and Arabic, and Swedish and Chinese). In general, we find that the pair of relevant classifiers tends to focus on the same spot, likely since multilingual text are often co-located with English. In the English and Arabic example, where the English and Arabic text are not co-located, we see that the English classifier focuses on the part containing English text and the Arabic classifier focuses on the part containing Arabic text, which suggests that each classifier focuses on the correct target language.

VI. APPLICATIONS TO STOCKHOLM

A. Summary Statistics

We apply our classifier on GSV images in Stockholm, Sweden between 2009 and 2021. We extract GSV images from Stockholm using the same method described in Section III-B. This amounts to 1026960 unique images, corresponding to 256740 unique panoramas. Since Google only conducts a large-scale update of images approximately every two years, we group the data in sets of two years (skipping 2015 as there is no data for 2015).

To facilitate comparison with public socioeconomic data from Statistics Sweden we aggregate our data at the DeSO (Demographic Statistical Areas) level. In Figure 6, we present choropleth maps of the linguistic concentration in Stockholm in 2020-2021.

We see that Swedish (unsurprisingly) has the strongest presence among the four languages across the city. English has a moderate presence, while Arabic and Chinese has minimal presence, and these three languages are more concentrated around the downtown area. As Hult observed in Sweden, English is not imposed from above but arises from socioeconomic interests [24]. It is therefore unsurprising to find a stronger presence of English in the downtown area where there are stronger commercial interests and higher tourist footfall.

[7]Modified implementation of: https://github.com/eclique/pytorch-Grad-CAM

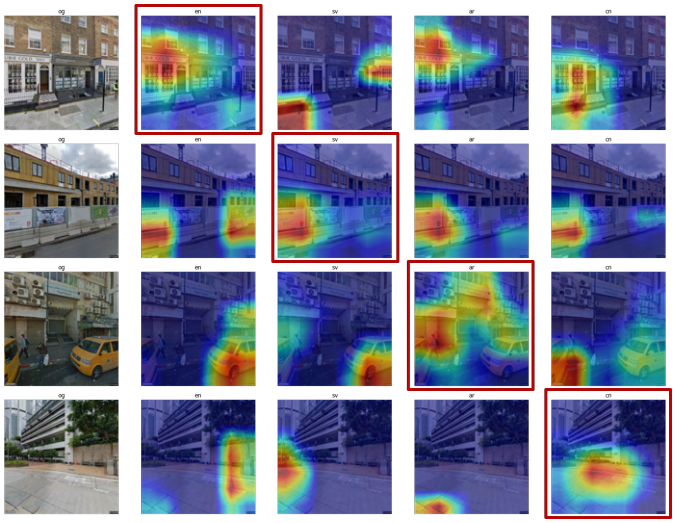

Fig. 3. Grad-CAM performed for true positive examples. There is one example for English, Swedish, Arabic and Chinese (from top to bottom). Heatmaps for each classifier (English, Swedish, Arabic and Chinese, from left to right) is provided for each image. For each row, the heatmap highlighted in red is that of the classifier for which the image is a true positive example. In general, we see that the model is attentive to parts of the images for which there is text e.g. store signs or road markings.

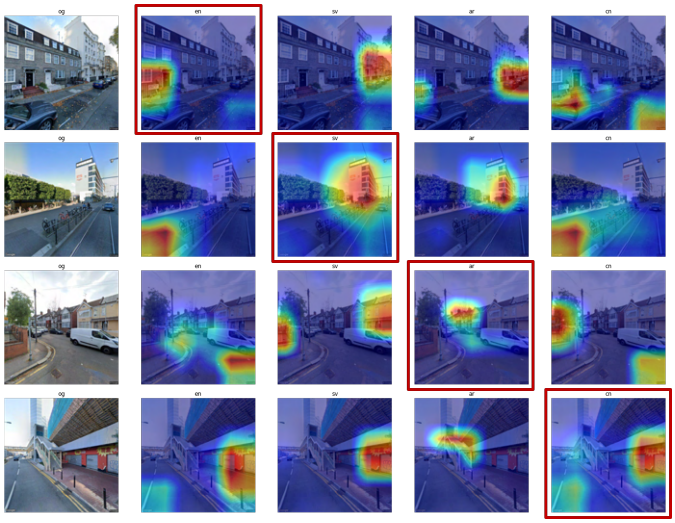

Fig. 4. Grad-CAM performed for false negative examples. There is one example for English, Swedish, Arabic and Chinese (from top to bottom). Heatmaps for each classifier (English, Swedish, Arabic and Chinese, from left to right) is provided for each image. For each row, the heatmap highlighted in red is that of the classifier for which the image is a false negative example. In general, we see that the model fails to detect the presence of text despite being attentive to parts of the images for which there is text. We postulate that this is because the text is too small or indistinct from other visual features in a streetscape.

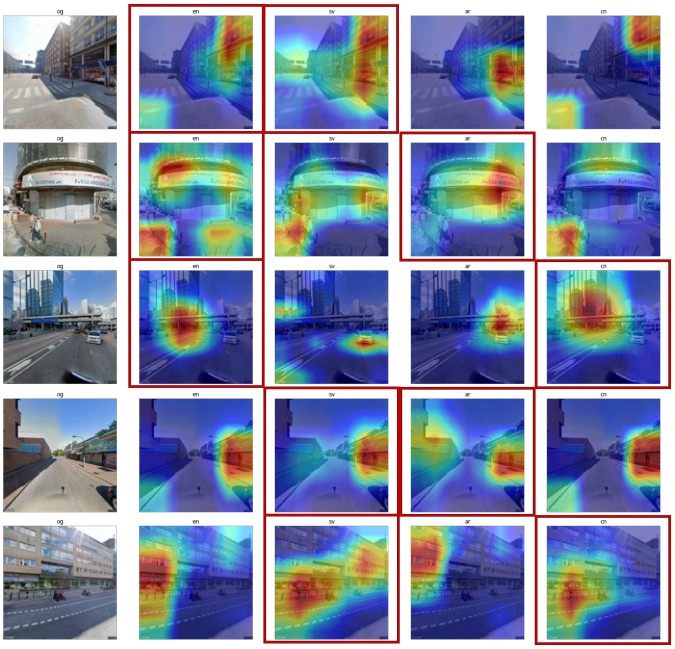

Fig. 5. Grad-CAM performed for true positive multilingual examples. There is one example for English and Swedish, English and Arabic, English and Chinese, Swedish and Arabic, and Swedish and Chinese (from top to bottom). Heatmaps for each classifier (English, Swedish, Arabic and Chinese, from left to right) is provided for each image. For each row, the heatmaps highlighted in red are those of the classifiers for which the image is a true positive example. In general, we see that the model is attentive to parts of the images for which there is text e.g. store signs or road markings. In multilingual scenes, each pair of classifiers tends to focus on similar parts of the image.

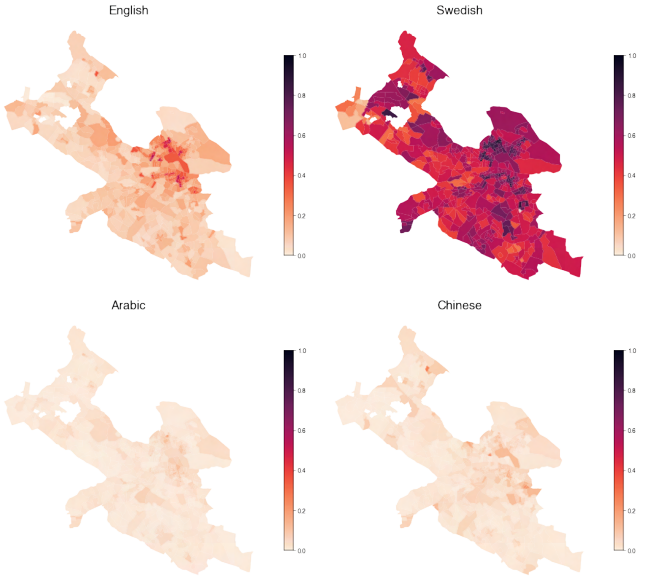

Fig. 6. Linguistic concentration of English, Swedish, Arabic and Chinese in Stockholm in 2020-2021. The presence of Swedish is much higher than all other languages across the city. There is moderate presence of English in the downtown area while the presence of other foreign languages is limited.

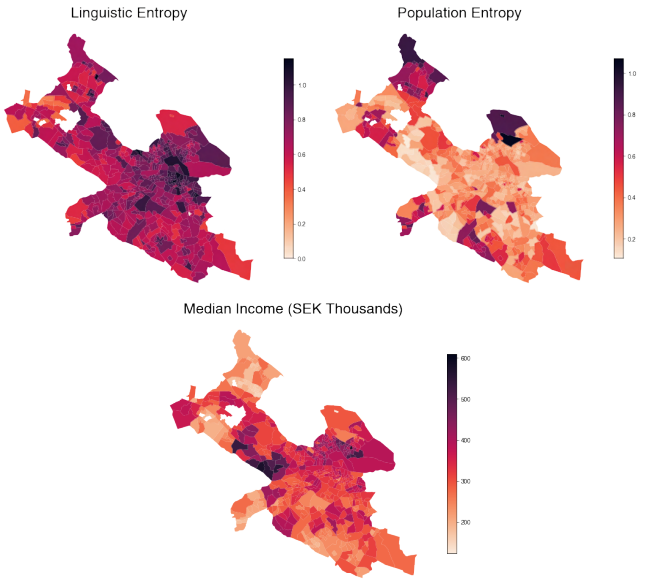

Fig. 7. Linguistic entropy, population entropy and median income in Stockholm in 2020-2021.

B. Comparison with other Socioeconomic Characteristics

Statistics Sweden provides aggregated data of the citizenship of residents—residents are classified as "Swedish", "Europeans except Swedish", or "Others". To facilitate comparison with our measures of linguistic diversity, we construct a measure of population entropy in the same way we constructed our measure of linguistic entropy. In Figure 7, we present choropleth maps of linguistic entropy and population entropy in Stockholm in 2020-2021. We also include a choropleth map of median income to facilitate comparison.

Interestingly, and perhaps counterintuitively, the spatial distributions of the two measures are reversed, with areas that have higher linguistic entropy having lower population entropy. In fact, the correlation of the two measures across all years is -0.205, implying a weak negative correlation. Although this might not make sense at face value, this observation aligns well with our understanding of Swedish society. Crucially, we need to recognize that high linguistic mix need not arise from high population mix, particularly in a country whose heritage and culture is hallmarked by homogeneity. Rather, the linguistic diversity we see in Stockholm is not so much resultant of a diverse resident population along ethnolinguistic lines, but likely a feature of globalization. In fact, in a study of linguistic landscapes in Seoul, Hong finds an increased prevalence of Chinese signs despite a relatively unchanged Chinese population and attributes this to "the recent popularity of Chinese food in the Korean society" [9], suggesting that linguistic diversity can be driven by preferences. One telltale sign for our case is the positive correlation between linguistic entropy and median income—0.261 which suggests that linguistic diversity may be well-associated with a diversity-seeking upper-middle class population that demands more culturally diverse goods and services.

On the other hand, despite the strong population entropy in the outskirts of the city, the lower linguistic entropy suggests

that minorities are not necessarily comfortable expressing themselves in their mother tongues, in a country that prides itself on its cultural homogeneity. In fact, Daun notes that differences in cultural backgrounds are downplayed in accordance with the Swedish "emphasis on conflict avoidance" [1], and this may have contributed to the limited presence of minority languages, similar to the case of Polish in Norway.

VII. Conclusion

In this paper, we introduce a different paradigm for detecting the presence of languages in streetscapes. Instead of using existing OCR tools, we propose the use of a pre-trained DenseNet-121 model to do binary classification for the presence of each language of interest. Our best model (which supports English, Swedish, Arabic and Chinese) achieves a test accuracy of 80.8%, surpassing the performance of existing OCR tools. We explore the use of both synthetic and real data in training our model and find that training solely on real data achieves the best performance, likely because synthetic data does not generalize well to real streetscapes and text in the wild. To check if the model is making sense, we employ Grad-CAM and find that the model is indeed focusing on visual features containing text. We apply our model to GSV images in Stockholm and find that the linguistic concentration of each of the four languages aligns with our intuition. We construct an entropy index to measure linguistic diversity and compare it with population entropy. Although there is a weak negative correlation between the two, we argue that linguistic mix and population mix need not go hand-in-hand, particularly in a strongly homogeneous society like Sweden. Rather, the fact that linguistic mix is not commensurate with population mix points to the importance of measuring them separately. The insights from our application in Stockholm highlights the potential of deploying language detection models in cities worldwide to obtain fine-grained characterizations of linguistic landscapes. Each characterization can help dig deeper into the state of social integration in each city, and comparing characterizations across cities with different cultural histories may generate further insights into the interaction between cultural history and observed diversity today.

References

[1] Å. Daun, "Swedish mentality," in *Swedish Mentality*. Penn State University Press, 2021.

[2] R. Andersson and A. Kährik, "Widening gaps: Segregation dynamics during two decades of economic and institutional change in Stockholm," in *Socio-Economic Segregation in European Capital Cities*. Routledge, 2015, pp. 134–155.

[3] R. Landry and R. Y. Bourhis, "Linguistic landscape and ethnolinguistic vitality: An empirical study," *Journal of Language and Social Psychology*, vol. 16, no. 1, pp. 23–49, 1997.

[4] D. Gorter, "Multilingual inequality in public spaces: towards an inclusive model of linguistic landscapes," *Multilingualism in the public space: Empowering and transforming communities*, 2021.

[5] L. Onofri, P. Nunes, J. Cenoz, D. Gorter *et al.*, "Linguistic diversity and preferences: Econometric evidence from European cities," *Journal of Economics and Econometrics*, vol. 56, no. 1, pp. 39–60, 2013.

[6] T. Opsahl, "Invisible presence? polish in norwegian public spaces," *Multilingualism in Public Spaces: Empowering and Transforming Communities*, p. 111, 2021.

[7] M. Barni and C. Bagna, "The critical turn in LL: New methodologies and new items in LL," *Linguistic Landscape*, vol. 1, no. 1-2, pp. 6–18, 2015.

[8] J. Leinonen *et al.*, "Researching in/visibility in the nordic context: Theoretical and empirical views," *Nordic Journal of Migration Research*, vol. 4, no. 4, p. 161, 2014.

[9] S.-Y. Hong, "Linguistic landscapes on street-level images," *ISPRS International Journal of Geo-Information*, vol. 9, no. 1, p. 57, 2020.

[10] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016. [Online]. Available: https://arxiv.org/abs/1608.06993

[14] K. Zhang, Y. Guo, X. Wang, J. Yuan, and Q. Ding, "Multiple feature reweight densenet for image classification," *IEEE Access*, vol. 7, pp. 9872–9880, 2019.

[15] J. Zhang, C. Lu, X. Li, H.-J. Kim, and J. Wang, "A full convolutional network based on densenet for remote sensing scene classification," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 3345–3367, 2019. [Online]. Available: https://www.aimspress.com/article/doi/10.3934/mbe.2019167

[16] H. Hassan, A. El-Mahdy, and M. E. Hussein, "Arabic scene text recognition in the deep learning era: Analysis on a novel dataset," *IEEE Access*, vol. 9, pp. 107 046–107 058, 2021.

[17] R. Ma, W. Wang, F. Zhang, K. Shim, and C. Ratti, "Typeface reveals spatial economical patterns," *Nature Scientific Reports*, vol. 9, no. 15946, 2019.

[18] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[19] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3752–3761.

[20] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[21] J. Amcoff, "Searching for new ways to achieve mixed neighbourhoods," *Cities*, vol. 121, no. 103496, 2022.

[22] R. Mora and J. Ruiz-Castillo, "Entropy-based segregation indices," *Sociological Methodology*, vol. 41, no. 1, pp. 159–194, 2011.

[23] Y. Song, L. Merlin, and D. Rodriguez, "Comparing measures of urban land use mix," *Computers, Environment and Urban Systems*, vol. 42, pp. 1–13, 2013.

[24] F. M. Hult, "English on the streets of Sweden: An ecolinguistic view of two cities and a language policy," *Working Papers in Educational Linguistics (WPEL)*, vol. 19, no. 1, p. 3, 2003.