DESCRIBING URBAN EVENTS USING DEEP LEARNING TECHNIQUES

Tales Lima Fonseca

Lucas Arantes Berg

Leonardo Goliatt

Marcos de Mendonça Passini

taleslimaf@gmail.com

berg@ice.ufjf.br

goliatt@gmail.com

mmpassini@ice.ufjf.br

Federal University of Juiz de Fora

Rua Jose Lourenço Kelmer, S/n - Martelos, 36036-330, Minas Gerais, Juiz de Fora

Abstract. *With the development of new information science technologies, large amounts of data are being generated every day in the urban environment, making it necessary to use computational techniques for its processing and analysis. In some cases the analysis depends on a human agent to interpret and evaluate a given situation. When devices generate data continuously, the amount of information that must be processed is limited by the performance of the worker and can result in a potential loss of quality in interpreting tasks. Considering this aspect, the use of intelligent devices can contribute considerably the improvement of the services of the traffic management and monitoring bodies. One example is intelligent traffic lights that identify vehicle flow through sensors and define how long they should be open in order to minimize local traffic jams or even to classify an image based on the object or in a certain situation. Artificial intelligence techniques have been shown to be efficient in this type of task, because they have the ability to analyze large datasets and make decisions in a short period of time. This article presents a tool that implements the concepts of Smart Cities and Artificial Intelligence in public data from the organs of traffic management of the city of Juiz de Fora. A deep learning neural network was developed to analyze data from monitoring cameras for traffic events.*

Keywords: *Smart Cities, Deep Learning, Traffic, Cameras*

1 INTRODUCTION

The concept of Smart Cities are diverse, covering different definitions depending on the meaning of the word "smart": Digital City, Wired City, Knowledge City, Green City, Virtual City and etc (Cocchia, 2014). There are many definitions of Smart City, but none of them have been universally recognized yet (Nam and Pardo, 2011). Washburn et al. (2009) conceptualizes an Smart City, placing an explicit emphasis on the use of intelligent computational technologies. They considered that the current urban crises served as a good pretext for an Smart City initiative. Some governments and government agencies are embracing the notion of intelligence to characterize their new policies, strategies and programs to achieve sustainable development, strong economic growth and better quality of life for their citizens (Centre On Governance, 2003).

The analysis, connection and integration of Information and Communications Technology (ICT) present in the urban environment provides a more consistent and intelligent understanding of the city, improving the efficiency and sustainability of the city (Hancke et al., 2013). In addition, these data can be used to better describe, model and predict urban behaviors and may make possible the simulations of impact forecasts of an urban development project (Batty et al., 2012).

It is important to note that the integration of ICT into urban infrastructure is not what makes a city smart (Hollands, 2008), but rather like it together with social and human capital, and a broad economic policy is used to drive increase and management of urban development (Caragliu et al., 2009). The data that are generated by ICT are seen as an essential content for the ideology of an Smart City.

Studies focused on Smart Cities are becoming more and more present. Several approaches are proposed for different urban problems, such as:

- Use of an event-oriented architecture for the monitoring of public spaces with heterogeneous sensors, improving detection of abnormal events and simplifying tasks and communications (Filipponi et al., 2010).
- Development of an alarm system that uses intelligent traffic lights to inform drivers about traffic and weather conditions (Barba et al., 2012).
- Design of an architecture for Smart Cities, where anyone can have access to real time data which has been collected using various sensory mechanisms. Helping the analysys and decision-making for future planning (Jalali et al., 2015).
- Creation of a routing protocol that is able to minimize power consumption while maximizing the useful life of a wireless network, enabling low-cost distributed monitoring systems that are the basis of intelligent cities (Baniata and Hong, 2017).

Governments are using real-time analysis to manage aspects of how a given city works and whether it is regulated. Vehicle traffic on a transport network is a common example of this, where data for cameras and transponders are sent to a control center for traffic flow monitoring, traffic light set-up, speed limit and penalties for traffic violations (Dodge and Kitchin, 2007).

This work uses a public online system for viewing traffic cameras on the main roads of the city of Juiz de Fora to acquire a dataset of images that will be embedded on a deep neural network with the objective of describing the situation which is represented on the picture. This

was done by using *Python* scripts and open source libraries for neural network and computer vision.

2 MATERIALS AND METHODS

2.1 Traffic Images Dataset

In the city of Juiz de Fora, there are a public online system that provides to any user a way for viewing traffic cameras distributed on the main roads. In this system, there are 12 cameras distributed in the main roads of the city.

The data were collected between the day 2017/23/08 at 16:40:10 until 2017/24/08 at 07:08:04 on the crossing of the avenues Rio Branco and Presidente Itamar Franco, one of the main roads of the city. This region is very bustling in certain periods of the day, such as on the early hours of the morning and during the beginning of the evening, as we could see in Figure 1.

(a) In the morning. (b) In the beginning of the evening.

Figure 1: Examples of blurry images captured by the camera.

Within the time the data were acquired it was able to extract a total of 177 videos, this is because the system is updated in intervals of five minutes and also each video from the monitoring service has a length of approximately one minute. In order to analyze traffic events, we implement a simple script based on the command *ffmpeg*. From that we extract each frame of the videos accounting a total of 83899 frames.

2.2 Pre-processing step

One major concern about the traffic images is related to blurred effects caused by the movement of the camera. Also, for the particular place we are taking our data, there is a residential area which is unavailable, so when the camera focus this area it covers using a blue rectangle in order to avoid showing it in public.

In order to eliminate these frames we needed a way to detect blurry images. Blurry detection could be made by computing the focus level of each pixel from the image. The work of (Pertuz et al., 2013) compares 36 different methods related to focus measumerents, one of them is the Variance of the Laplacian. The method is described in (Pech-Pacheco et al., 2000) and

the basic idea is first convert the target image to grayscale and apply the following kernel on that channel by using a convolution operation:

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

After that we calculate the variance of the result and check if this value is below a predefined threshold. If this is the case we classify the image as blurry, otherwise, the image is not blurry. Figure 2 shows an example of classification of different images with threshold equal to 100.0.

(a) Not Blurry ($Variance = 388.85$) (b) Blurry ($Variance = 2.20$)

Figure 2: Example of images not blurry and blurry, respectively.

The *OpenCV* (Itseez, 2015) is an open source computer vision and machine learning library, which is widely used in many project related to image processing. The library provide optimized algorithms for a lot of different image-related tasks, such as the Variance of the Laplacian filter, which was used in this project by implementing a *Python* script.

2.3 Deep neural network

In this article, we used the "Show and Tell" model (Vinyals et al., 2017) to obtain a description of the content present from each image analyzed. This model is an example of an encoder-decoder neural network that works by first "encoding" an image into a fixed-length vector representation, and then "decoding" the representation into a natural language description.

The process of "encoding" an image of the model is a deep Convolutional Neural Network (CNN) (LeCun et al., 1989). CNN is a type of network that is extensively used for image tasks and beyond that, is currently state-of-the-art for object recognition and detection. This model

uses a particular CNN, the *Inception v3* (Szegedy et al., 2016). *Inception v3* is a image recognition model that was pretrained on the ImageNet (Russakovsky et al., 2012) image classification dataset.

The process of "decoding" the representation is a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), a recurrent neural network architecture that is able to remembers values over arbitrary intervals. LSTM is a type of network that is frequently used for language modeling, machine translation, question answering and etc. In this model, the LSTM network is trained as a language model conditioned on the image encoding.

An embedding model (Mikolov et al., 2013) was used to represent the words in the descriptions. In this way, each word in the vocabulary is associated with a fixed-length vector representation that is learned during training. In this model, we used 512 dimensions for the embeddings and for the size of the LSTM memory.

Figure 3 shows a diagram that illustrates the model architecture.

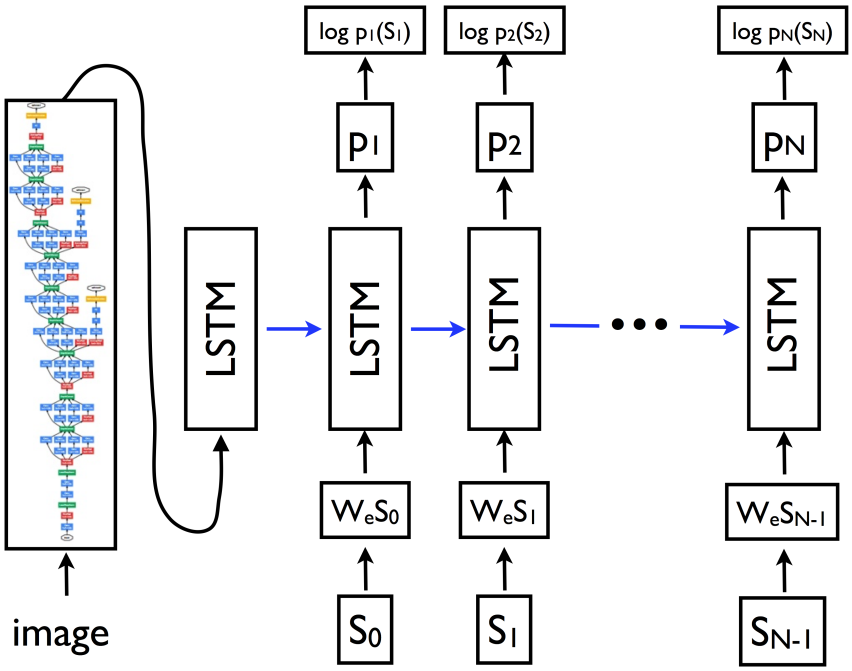

Figure 3: LSTM model combined with a CNN image embedder and word embeddings.

In the "Show and Tell" model architecture, the set of variables $\{s_0, s_1, \ldots, s_{N-1}\}$ represents the words of the description and $\{w_e s_0, w_e s_1, \ldots, w_e s_{N-1}\}$ are their corresponding word embedding vectors. The LSTM outputs, represents by $\{p_1, p_2, \ldots, p_N\}$, are probability distributions that was generated by the model to predict the next word in the sentence. The values $\{\log p_1(s_1), \log p_2(s_2), \ldots, \log p_N(s_N)\}$ are the log-likelihoods of the correct word at each step of the training and the negated sum of these values is the minimization objective of the model.

The training consists of two phases. In the first phase, the parameters of the *Inception v3* model are kept fixed and it's used as a static image encoder function. On the top of the *Inception v3* model, a single trainable layer is added in order to transform the image embedding into the word embedding vector space. So, with that, the "Show and Tell" model trains the parameters of the word embeddings, the layer on top of *Inception v3* and the LSTM. In the second phase, all

parameters of the model are trained simultaneously for better adjustment of the image encoder and the LSTM.

3 RESULTS AND DISCUSSION

To train the "Show and Tell" model, we used the MSCOCO (Lin et al., 2014) image captioning dataset. This dataset contains images and their respective description, where the description is a list of words. In the preprocess of this dataset, a dictionary is created to assign each word present in the vocabulary of the descriptions to an integer value. In this way, after the preprocess, each description is encoded as a list of integers.

After training the "Show and Tell" model on the MSCOCO image captioning dataset, is able to generate a completely new caption using concepts learned from similar scenes in the training set. Figure 4 shows three images of the training set along with its caption and the caption generated for a new image that don't belong to the training set.

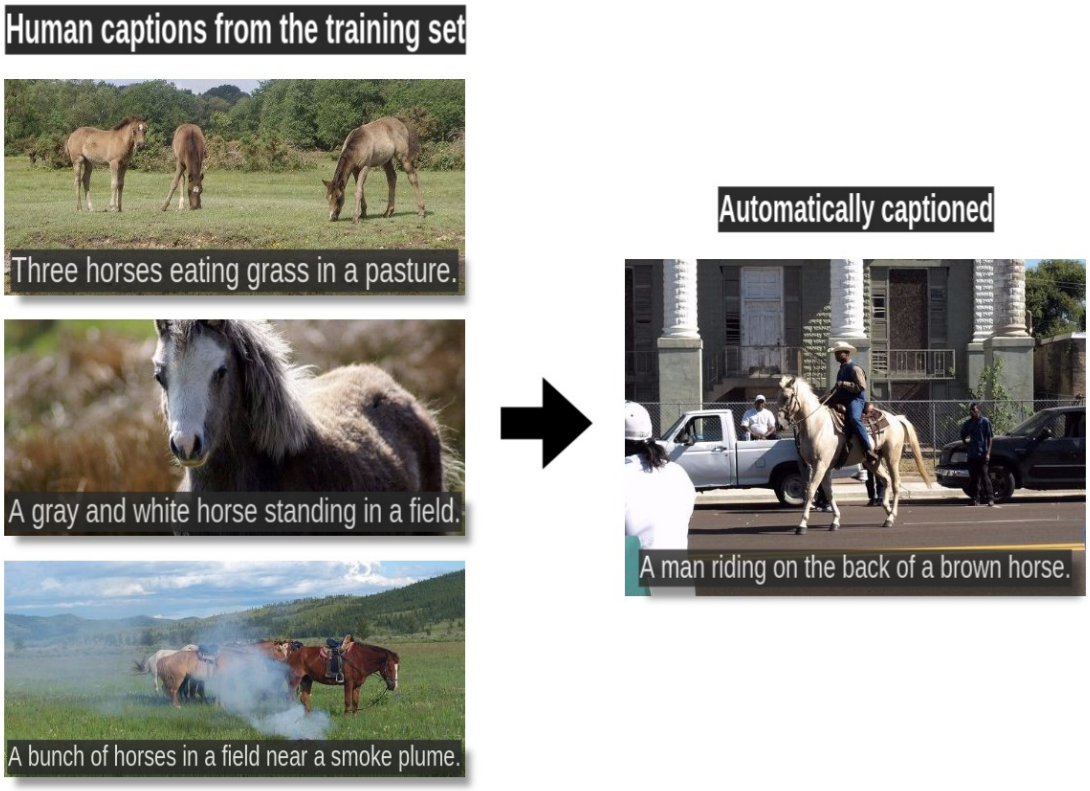

Figure 4: Example of the captioning process on the trained "Show and Tell" model with MSCOCO image captioning dataset.

Using the Laplacian filter it was possible to eliminate the frames that capture the movement of the camera with excessive amounts of blurring and also some frames that show residential areas that are blocked by the camera with a blue rectangle, as shown in Figure 5. We choose a threshold of 2500 for the identification of a blurry image, as a result we reduce the ammount of frames to 25223.

After this preprocessing step and the training of the "Show and Tell" model on the MSCOCO dataset, we apply each of the images in the reduce set of Traffic Images Dataset on the model,

(a) Blocked residential area (b) Blurry Street

Figure 5: Examples of blurry images captured by the camera and that were eliminated after the pre-processing step.

and we obtain a description of the content from each image. Figure 6 presents a flowchart of the proposal of this work and Figures 7 and 8 shows some caption examples generated by the model.

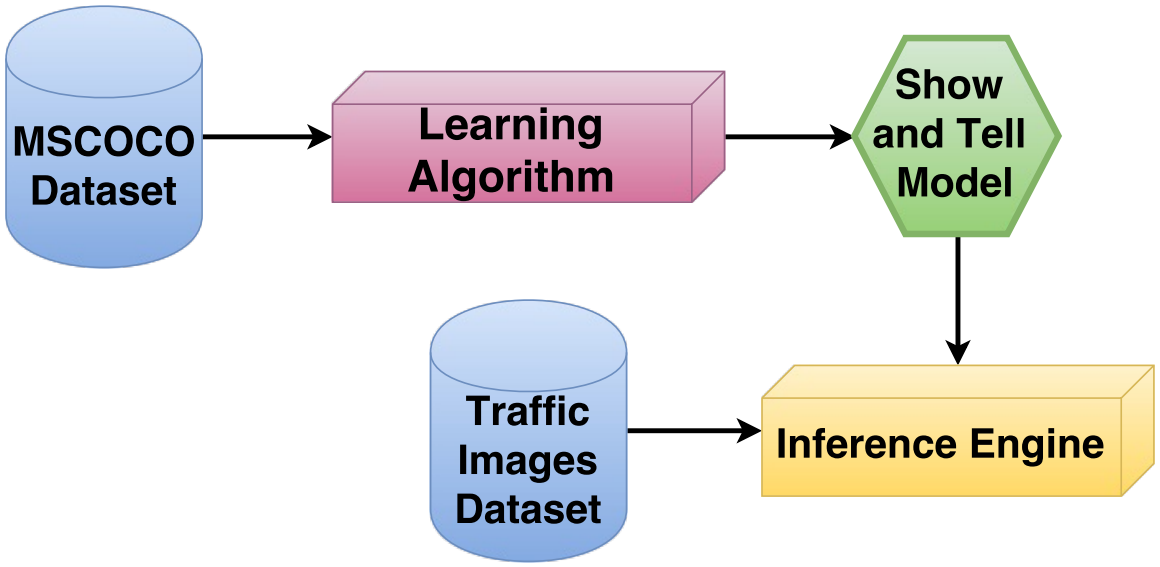

Figure 6: Flowchart of the proposal of this work.

From the Figures 7 we could see that the trained model could identify important aspects from the images. For example in Figure 7 (a) the trained model classify the situation from the image as a public transit bus on a city street, which is precisely correct. One of the reasons for the probability of this classification was so high is that on the figure there are more than one bus and also in this particular period of the day there was not too much people walking on the street, fact that could potentially confuse the trained model and lead to other conclusions.

Another aspect of the classification from the trained model that was observed from the results is that when there is not too much information on the image the trained model gave us better results. For example, if we check Figure 7 (b) there is nothing around the motorcyclists, so the trained model could easily match the concept of what is a bike and return a correct

(a) A public transit bus on a city street. ($p = 0.018445$)

(b) A group of people riding bikes down a street. ($p = 0.010015$)

(c) A city street filled with lots of traffic. ($p = 0.009102$)

(d) A city street filled with lots of traffic. ($p = 0.00898$)

Figure 7: Description of the content from the images with the highest probability from the output of the neural network.

description of the situation.

When the trained model faces a image with a lot of information, like in Figures 7 (c) and 7 (d), the output should be as much general as possible in order to get a overview of the situation. In these images in particular, the trained model could identify a great number of cars, the semaphore and some signs on the street, by linking all of these elements together the output was a city filled with lots of traffic, which is in fact a correct and general description of what is going on.

But we note that the trained model used has its limitations. The worst results were obtained mostly at night periods and when it was not possible or to difficult to identify key aspects of the images. For example, in Figure 8 (a) we could describe this image as a street at night filled with lots of traffic, but instead the trained model described as a view of a train station with a train in the background. One possibility that could cause this error is that all the cars have the same color and the way that the two last cars were aligned on this image may look like the shape of the back of a train.

Another example of this misunderstanding occur in Figures 8 (b) and 8 (c). On Figure 8 (b) the output of the trained model was a view of a city with a bus and a building, but no bus is on

(a) A view of a train station with a train in the background. ($p = 0.0$)

(b) A view of a city with a bus and a building. ($p = 0.0$)

(c) A truck is driving down the street in the city. ($p = 0.0$)

(d) A view of a city at night with people walking around. ($p = 0.0$)

Figure 8: Description of the content from the images with the lowest probability from the output of the neural network.

the image. Instead the trained model might thought that the two cars waiting at the traffic light was actually a bus, this is because from the point of the camera the two cars were quite alligned to each other resembling the shape of a mini bus. On Figure 8 (c) we got a truck driving down the street in the city, but analyzing the image we could conclude that the trained model confused a pickup with a truck.

Lastly, on Figure 8 (d) there is another type of error that can occur using this trained model. On the image we could see a street at night with a lot traffic, but the trained model instead describe the image as a view of a city at night with people walking around. With that observation we could conclude that the trained model gives more attention to what is on the middle of the image, since there is only people on the center and much of the traffic is on the background of the picture.

4 CONCLUSIONS

This work presented an approach to describe city events captured by a monitoring camera using a deep neural network. The results of the descriptions in most cases were relatively close to what a human could describe. From the results was possible to verify that the trained model

from the network is able to match common street objects, like cars, bikes, buses, signals and people, even with a general training compose by other types of situations, which was not related to city streets.

Some improvements could be made by training the network with specific situations which are all related to city streets and traffic jams, this would increase the probabilities of an situation be taken from the trained model and lead to better results. Also it was possible to notice that in some situations the trained model returned an incorrect description, which in most cases was because the appearance of similar objects to the ones the trained model know, like the pickup being confused to a truck in Figure 8 (c). This behaviour happened when these objects were on the foreground of the picture, so a more general reading of the objects on the picture and also a great number of different objects related to city streets could help the trained model in this distinction.

MSCOCO is a large image dataset designed for object detection, segmentation, and caption generation. Nevertheless, this database presents a large amount of information that is not relevant to traffic analysis, as well as, food, electronics, furniture, animals and others. Due to this, the intervention of human-network cooperation, such as transit agents, in the process of make a new dataset for the transit analysis would also improve the performance of the network to describe images from the cameras.

The availability of videos in a continuous and better quality in the city of Juiz de Fora, together with a specific database for traffic analysis, would make possible the creation of a constant monitoring system of roads.

ACKNOWLEDGEMENTS

The authors wish to thank CNPq, FAPEMIG, CAPES and the Graduate Program in Computational Modeling of the Federal University of Juiz de Fora for their support.

REFERENCES

Baniata, M. and Hong, J. (2017). Energy-efficient unequal chain length clustering for wireless sensor networks in smart cities. *Wireless Communications and Mobile Computing*.

Barba, C. T., Mateos, M. A., Soto, P. R., Mezher, A. M., and Igartua, M. A. (2012). Smart city for vanets using warning messages, traffic statistics and intelligent traffic lights. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 902–907. IEEE.

Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., and Portugali, Y. (2012). Smart cities of the future. *The European Physical Journal Special Topics*, 214(1):481–518.

Caragliu, A., Del Bo, C., and Nijkamp, P. (2009). Smart cities in europe, series research memoranda 0048. *VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics*.

Centre On Governance (2003). Smartcapital evaluation guidelines report: Performance measurement and assessment of smartcapital. https://goo.gl/vQaRS1. University of Ottawa.

Cocchia, A. (2014). Smart and digital city: A systematic literature review. In *Smart city*, pages 13–43. Springer.

Dodge, M. and Kitchin, R. (2007). The automatic management of drivers and driving spaces. *Geoforum*, 38(2):264–275.

Filipponi, L., Vitaletti, A., Landi, G., Memeo, V., Laura, G., and Pucci, P. (2010). Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors. In *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pages 281–286. IEEE.

Hancke, G. P., Silva, B. d. C. e., and Hancke, Jr., G. P. (2013). The role of advanced sensing in smart cities. *Sensors*, 13(1):393–425.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hollands, R. G. (2008). Will the real smart city please stand up? intelligent, progressive or entrepreneurial? *City*, 12(3):303–320.

Itseez (2015). Open source computer vision library. `https://github.com/itseez/opencv`.

Jalali, R., El-Khatib, K., and McGregor, C. (2015). Smart city architecture for community level services through the internet of things. In *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, pages 108–113. IEEE.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nam, T. and Pardo, T. A. (2011). Conceptualizing smart city with dimensions of technology, people, and institutions. In *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times*, pages 282–291. ACM.

Pech-Pacheco, J. L., Cristóbal, G., Chamorro-Martinez, J., and Fernández-Valdivia, J. (2000). Diatom autofocusing in brightfield microscopy: a comparative study. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 314–317. IEEE.

Pertuz, S., Puig, D., and Garcia, M. A. (2013). Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2012). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2017). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.

Washburn, D., Sindhu, U., Balaouras, S., Dines, R. A., Hayes, N., and Nelson, L. E. (2009). Helping cios understand "smart city" initiatives. *Growth*, 17(2):1–17.