August 07, 2018

COLOR-BASED OPTICAL CAMERA COMMUNICATION USING DEEP LEARNING

Nicola Rohrseitz

COLOR-BASED OPTICAL CAMERA COMMUNICATION USING DEEP LEARNING

AUTHORS:
Nicola Rohrseitz

ABSTRACT

When a network is not available or allowed (e.g., air gap), all benefits of digital communication and the Internet disappear. To enable substantial data transmission, techniques are described herein for using Machine Learning to embed data into color images and to read it with camera-equipped devices. In addition to data transmission, this enables new applications such as embedding of content on packaging (instructions or runnable code and configurations directly on devices), physical documents with selective encryption (role-based permission to read parts of a document), protection from digital tracking, prevention of tampering on physical documents, and more.

DETAILED DESCRIPTION

Passing information to a device without cabled or radio-based wireless connectivity is an ongoing problem. One approach is to use a different bands of the radio spectrum, such as visible light. This is the approach taken in the fields of Visible Light Communication (VLC) and Optical Camera Communication (OCC). VLC has emerged as a promising technology for wireless communication owing to advantages such as bandwidth, license, coexistence, and security. However, VLC requires an active source (i.e., emitting light) to modulate the signals. On the other hand, OCC can also operate on passive sources, that is, data printed on pictures, such as Quick-Response (QR) codes.

A QR code® is a type of two-dimensional matrix barcode first designed for the automotive industry in Japan. A barcode is a machine-readable optical label that contains information, typically about the item to which it is attached. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and Kanji) to efficiently store data and consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as a camera and processed using Reed-Solomon error correction until the image can be appropriately interpreted. The required data is then extracted from patterns that are present in both horizontal and vertical components of the image. QR codes became popular outside the automotive industry due to their fast readability and greater storage capacity compared to standard Universal Product Code (UPC) barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing.

The amount of data a standard QR code can carry is limited by the number of its pixels. To increase the payload, the number of pixels needs to be increased: either by making the QR code bigger or by making the pixels smaller. Both options are constrained by the measurement devices: to read more data, either the field of view or the pixel resolution must be larger.

Another approach to increase the payload is to increase the amount of data per pixel. This can be accomplished by using more than the two black and white states, for instance greyscale and color. The more shades that can be robustly measured, the larger the payload.

Accurately measuring color requires robustness to variations due to incoming light. Without compensation, the intensity and the color of the light source modify what the photoreceptors ("pixels") of a camera can measure. This problem of "color constancy" remains largely unsolved by technical systems. Even though biological vision achieves color constancy, its precise mechanisms are not yet fully understood. Functionally, it is clear that there is an evolutionary advantage in maintaining colors constant in forests speckled by light or at different times of day.

To mitigate for this effect, current techniques use a reduced color palette, for instance four colors. Given the ubiquity of QR codes, the application potential of using more colors to increase bandwidth in optical wireless transmission information is vast. For one, when standard networking technologies are unavailable or not allowed (e.g., air gaps), the enormous societal and business benefits of digital communication and the Internet disappear. Accurately measuring color would enable networking using, for instance, QR codes, pictures, or electronic displays as data emitters and cameras or other measurement devices as receivers. Such a communication system also carries important technical benefits in terms of latency and availability.

The techniques described herein use a deep neural network trained specifically to measure color in the context of data transmission. Although some current techniques relevant to OCC make use of neural networks, their role is not to measure color. Empirical results indicate very high accuracy (98%) in a task that had not yet been solved by a technical system.

The deep neural network is capable of accurately measuring the color of a region of the field of view by including in the processing the adjacent areas (in contrast, cameras are designed to convey only the measured color of one specific region, the "pixel"). A Machine Learning process prior to deployment determines which locations and calculations the neural network needs to perform.

To create the deep neural network capable of accurately measuring color, a workflow of three high-level steps is carried out: (1) creation of a relevant dataset, (2) training of the deep neural network, and (3) deployment on a camera-carrying device.

The first step involves acquiring a most complete and diverse dataset. To be robust against the main problem of accurately measuring color - namely changes in illumination - a deep neural network is trained on a dataset that contains multiple pictures of the colors to be recognized, each under different light intensities and colors. The dataset is stored with metadata that describe the characteristics to be learned.

In the second step, an appropriate deep learning architecture is designed and parametrized. It is typical of Machine Learning to have many variants of architecture and parameters that lead to a similarly effective systems. The dataset may also be different and still be effective. It is the practitioner's role to find the right combination to achieve the intended goals.

The third step involves transferring the learned model to a deployment system for inference (for instance, a smartphone or an industrial measurement system), whose camera characteristics must match those of the camera used for acquiring the dataset.

For experimental verification, a dataset was created, the neural network trained, and a measurement application implemented on a mobile device. To capture the dataset, an array of colored elements were placed under a camera, surrounded by a set of programmable color light bulbs. An application was run which at fixed time intervals (two seconds) automatically changes the values of hue, brightness, and saturation by predefined amounts, while capturing pictures with the camera. To capture the metadata, the standard procedure was followed of storing each picture file in a folder corresponding to the incoming light characteristics.

To train the neural network on the dataset, the following characteristics were used:

- Type: convolutional neural network
- Activation function: Rectified Linear Unit (ReLU); If input <= 0, then output = 0, otherwise if input >0, then output = input)
- Optimizer: Adam (ADAptive with Momentum - https://arxiv.org/abs/1412.6980)
- Loss function: categorical crossentropy (quantifies the difference between two probability distributions – http://deeplearning.net/software/theano/library/tensor/nnet/nnet.html#theano.tensor.nnet.nnet.categorical_crossentropy)
- Batch size: 128

The following code shows the details of the neural network model. It is written in Python and uses the Keras Application Programming Interface (API) to Tensorflow, which is an open-source development framework created by Google, Inc. (https://www.tensorflow.org).

```
model.add(Convolution2D(256, 1, 1, border_mode='same', input_shape=(3, 32, 32)))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dense(num_classes, activation='softmax'))
```

The trained neural network was ported to an application running on an iPad®, proving that the system can run on off-the-shelf hardware.

Figures 1-3 below illustrate a camera identifying a two-dimensional multi-colored matrix designated as number “20.”

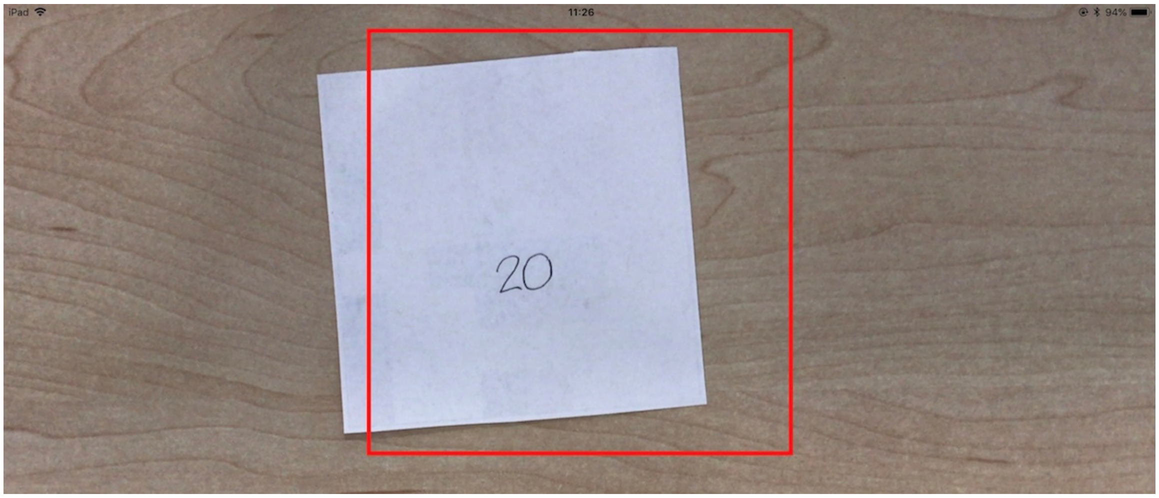

Figure 1

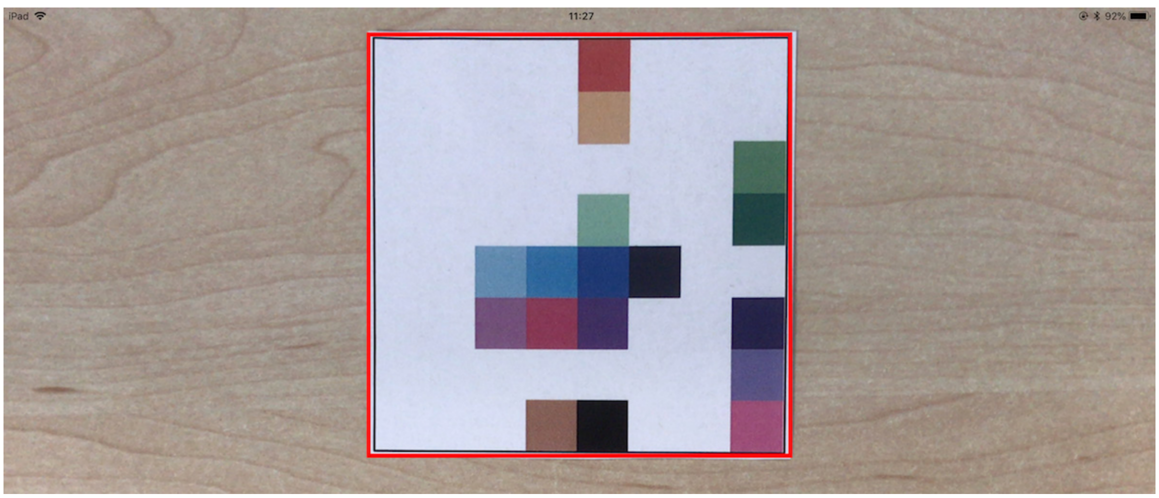

Figure 2

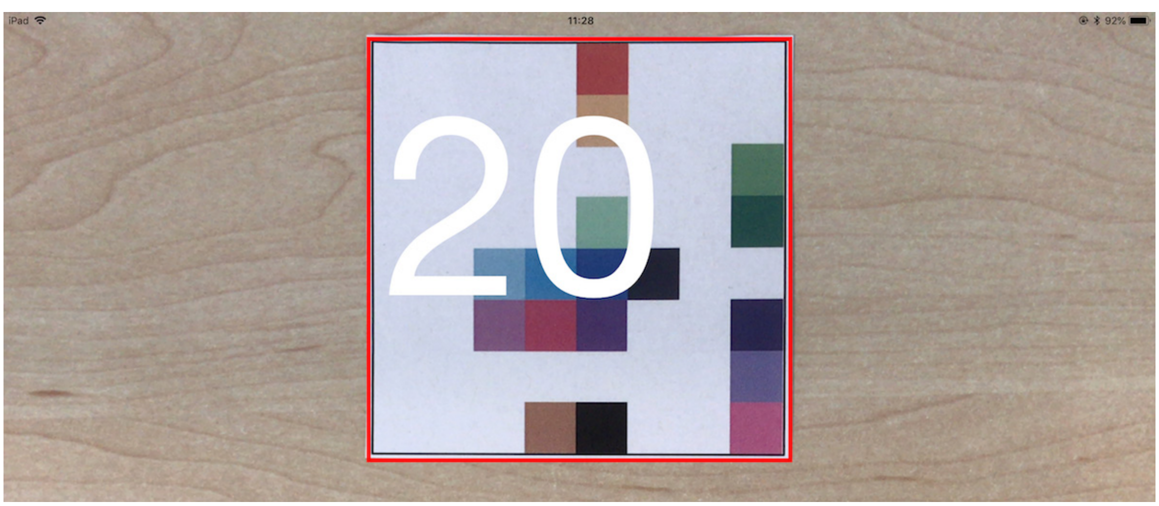

Figure 3

In summary, to enable substantial data transmission, techniques are described herein for using Machine Learning to embed data into color images and to read it with camera-equipped devices. In addition to data transmission, this enables new applications such as embedding of content on packaging (instructions or runnable code and configurations directly on devices), physical documents with selective encryption (role-based permission to read parts of a document), protection from digital tracking, prevention of tampering on physical documents, and more.