Expiry Date Digit Recognition using Convolutional Neural Network

Tareq Khan

Abstract — The expiry dates printed on the merchandise have a distinct background, font, alignment, and color in comparison with the available handwritten digit datasets. In this paper, an expiry date dataset is used, and also a convolutional neural network (CNN) model is proposed to recognize expiry dates out of images. This model may be employed together with our previously proposed smart expiry architecture to get an automated notification to the smartphone for the foods which are expiring soon. The suggested deep learning model is tested and has a classification accuracy of 90%.

Index Terms — convolutional neural network, dense layer, optimizer, dropout, expiry date.

I. INTRODUCTION

Annually, an estimated 1.3 billion tons of food is wasted globally [1]. One of the explanations for this food waste is that the customers fail to remember the expiry date of the food after purchasing and do not swallow the food earlier than its expiration. After the customers realize that the food passed beyond the expiration date, it's thrown out. Moreover, consuming expired food might cause food poisoning, fever, vomiting, dehydration, nausea, and diarrhea. To solve this issue, we suggested a cloud-based smart expiry system in [2]-[4] which sends an automated notification to the client's smartphone several days prior to the expiration date of the food.

English letters are utilized to compose the expiry date on an item. However, as we suggested in [2]-[4], the expiry date needs to be written using a barcode on the item. After the customer chooses the product and goes for check-out, the barcode of the item, in addition to the barcode for the expiry date, is scanned by the checkout operator. To implement this process, grocery stores are needed to print the barcodes of the expiry dates and then attach them to every item by hand. This can be an overhead for the shop owner. To resolve this issue, a convolutional neural network (CNN) based deep learning model is proposed in this paper to automatically identify the expiry dates out of pictures. This work can help to omit the task of manually placing the expiry date barcodes on each item. An image of the expiration date can be taken by the check-out operator and recognition of the date digits can be done automatically.

For recognizing digits and characters, optical character recognition (OCR) applications like Tesseract [5] can be utilized. However, due to usage of the various fonts,

backgrounds, colors on the product, lack of fully vertical or horizontal orientation of the digits – this OCR library has been unsuccessful to recognize the majority of the expiration date digits. MNIST dataset [6] includes a training group of 60,000 samples and a test group of 10,000 samples of handwritten digits pictures. This dataset cannot be utilized to train our deep learning model as it does not utilize the fonts and background used in the expiry dates. Our previous work in [7] used a fully connected (FC) neural network. In this paper, a CNN based deep learning model has been proposed to recognize the expiry date digits, which reveals better accuracy compared to our work in [7]. The proposed model is trained, validated, and tested with data that is concealed.

II. METHODOLOGY

A. The Dataset

We created a dataset of 1000 pictures of expiration date specimens and utilized them within this work [8]. The dataset includes 10 types of digits from 0 to 9. Every digit includes 100 images. The pictures were gathered by heading into different grocery shops and shooting hundreds of photographs of the expiration dates. Afterward, the digit pictures were resized and cropped to 32×32 pixels. They have three color channels - red, green, and blue. A number of these sample pictures are shown in Fig. 1. Here we see the pictures contain fonts that are usually not seen in text documentations like in Fig. 1 (a), a few pictures aren't completely vertical or horizontal as shown in Fig. 1 (b), plus they feature background colours as shown in Fig. 1 (g) to Fig. 1 (i) These pictures can't be comprehended by present OCR libraries [4].

Fig. 1. Examples of expiry date digit pictures in the dataset.

B. Convolutional Neural Network Architecture

To classify the expiry date digit pictures, a CNN based deep learning neural network, as revealed in Fig. 2, is used. Below is a brief description of different layers and optimizer.

Input Image: 3-D tensor of size (32, 32, 3) is used as the input image – containing separate channels for red, green, and blue. The pixel data type is changed from integer to floating-point. To make the pixel values in the range of 0 to 1 for normalization, they are divided by 255.

Submitted on November 29, 2020.
Published on February 24, 2021.
Tareq Khan, Eastern Michigan University, USA.
(corresponding e-mail: tareq.khan@emich.edu)

1) Convolution Layer:

This is a 2-D layer that utilizes sliding convolutional filters on the input image. The filters are moved vertically and horizontally. The dot product of the filter's weights along with the input image pixel values is calculated. Then a bias term is added [9]. In the proposed model, the filter dimensions are 3×3 and six convolutional layers are utilized. The filter values are learnable parameters and they have been all initialized with arbitrary values. At Fig. 2, the very first *conv2d* layer, we find 32 filters of dimension 3×3 using padding – so they create 32 output layers using the identical width and height of the input. In the proposed model: *conv2d*, *conv2d_2*, and *conv2d_4* use padding to generate the output size as like the input; however, padding is not utilized in *conv2d_1*, *conv2d_3*, and *conv2d_5*.

2) *Activation Layer*: The rectified linear unit (ReLU) [10] is used as the activation layer. This layer comes after each convolutional layer and after the dense layers (except the last dense layer). ReLU is a nonlinear activation function that converts any negative number to zero and keeps the positive numbers unchanged.

3) *Max Pooling Layer*: This layer is responsible for down-sampling. It divides the input into rectangular pooling areas, and sample only the largest value of every area [11]. A pooling rectangle of 2×2 is utilized in this model.

4) *Dropout Layer*: This layer takes a probability value as the argument and arbitrarily sets the input layer values to zero according to the specified probability argument. This aids to avoid the system from overfitting [12]. There is no learning parameter in this layer. In the suggested model, three dropout layers are used having a probability of 0.25. They help to solve the overfitting problem of the training data in a small number of epochs.

5) *Flatten Layer*: This layer converts the multidimensional input layer to a single column tensor. A one-dimensional vector of size 512 is generated in this model by converting the (2, 2, 128) input layer.

6) *Dense Layer*: This is the fully attached (FC) layer. The dot product of the input and a weight matrix is calculated in this layer. Subsequently, a bias vector [13][14] is added. The weight matrix and bias are learnable parameters. Before the start of training, the weight matrix and the bias are filled with arbitrary values.

Loss Function and Optimizer: A matching score – the more the match, the less the score - between the forecasted scores and the fact labels are calculated by the loss function. The optimizer's job is to reach the global minima of the loss function by adjusting the network parameters (i.e., filters, weights matrix, and biases) during the training time. To classify the image, the final fully connected layer, *dense_2*, aggregates all the features. The output dimension of this last dense layer corresponds to the total classes from the data set. As the dataset has 10 classes - from digit 0 to 9, the output dimension is set to 10. After this layer, the softmax [15] function and cross-entropy loss is used to find the matching score. The RMSprop optimizer [16] is used to train the model. 1×10^{-5} is used as the learning rate and a total of 1,077,290 trainable parameters are present in the proposed model.

Fig. 2. The proposed CNN architecture.

III. RESULT

Random shuffling was done on the dataset maintaining the ground fact label connections of the 1000 expiry date digit pictures. Afterward, the dataset is broken up into 3 groups: for training – 600 images, for validation – 200 images, and for testing – 200 images. Prior to the model is trained and verified, the 200 images for testing were retained hidden.

These isolated hidden images were then used to find the final accuracy of the model.

Python programming language with the *Keras* library was used to construct the deep learning model. Keras offers consistent and high-level application interfaces (APIs) for training and inferencing deep learning models. Keras is recently integrated with TensorFlow [17]. A computer with Intel Core i7 – 4.6 GHz processor, 32 GB random access memory, and NVIDIA GeForce GTX1060 graphics processing unit (GPU) was used to train the model.

Training and validation of the model were done concurrently for 50,000 epochs. It took about 1 hour 7 minutes to train and validate the model. In the plot in Fig. 3, loss vs epoch is shown for both training and validation datasets. In the plot in Fig. 4, accuracy vs epoch is shown for both training and validation datasets. The training loss and validation loss after 50,000 epochs are 0.07 and 0.84 namely as we see in Fig. 3. We see from Fig. 4 that accuracy improves as epochs increases. The training accuracy reached 0.98 and a validation accuracy reached 0.83 after the end of 50,000 epochs.

Fig. 3 Plot of loss vs epochs.

Fig. 4. Plot of accuracy vs epochs.

We notice from the above plots that validation loss grows after somewhere around 34,000 epochs, but validation accuracy does not go down, rather it continues to raise. The reason for this particular happening is that validation loss raises for a few marginal samples, but the increase of these losses hasn't crossed the threshold to be classified as another class. The more the epoch grows, the classes are forecasted

more accurately for other samples, and it causes the overall validation accuracy to go up.

The model parameters were saved following the training and validation and it has a size of 8.29 MB. Subsequently, the isolated test set of 200 images were used to test the model. The loss and accuracy of the test set were 0.70 and 0.90 namely. The model produced better accuracy for the test set compared with the validation set. This shows that good generalization is achieved. The time to recognize a single-digit image (i.e. the inference time) is found to be 1.12 seconds.

TABLE I. COMPARISON WITH OTHER WORK

Work	Model	Accuracy	Parameters
Our previous work [7]	FC	80%	23,667,594
Proposed work	CNN	90%	1,077,290

Table I shows the comparison with our previous work in [7]. Here we see that the proposed work uses CNN which works better for images than FC based neural networks. The proposed work has 10% more accuracy and the model has about 22 times a smaller number of learnable parameters. Thus, the proposed model will take lesser disk space.

IV. CONCLUSION

We have proposed a CNN based deep learning model in this work for classifying digits that are found on expiry dates. Training, validation, and testing of the model are done successfully. Our previously proposed smart-expiry architecture can utilize this model, which will help to eliminate the burden of attaching barcode labels manually for expiry dates. Increasing the size of the dataset and localization of the expiry date from the captured images are considered for future work.

ACKNOWLEDGMENT

The author would like to thank Mohammed Rashed Hyder for taking photographs and cropping the expiry date images.

REFERENCES

- [1] Food wastage: Key facts and figures, [Online]. Available: <http://www.fao.org/news/story/en/item/196402/icode/>, 2021.
- [2] T. H. Khan, "A cloud-based smart expiry system using QR code," *IEEE International Conference on Electro Information Technology (EIT 2018)*, pp. 245 – 248, Rochester, Michigan, USA, May 2018.
- [3] T. H. Khan, "Cloud-based architecture of a smart expiry system with IoT device", *International Journal of Advance Engineering and Research Development (IJAERD)*, vol. 5, no. 4, pp. 641-650, 2018.
- [4] M. R. Hyder and T. H. Khan, "Automatic expiry date notification system interfaced with smart speaker," *International Journal of Engineering Science Invention (IJESI)*, vol. 9, issue. 7, pp. 14 – 20, 2021.
- [5] Tesseract OCR, [Online]. Available: <https://opensource.google.com/projects/tesseract>, 2020.
- [6] The MNIST Database, [Online]. Available: <http://yann.lecun.com/exdb/mnist/>, 2021.
- [7] T. H. Khan, "Expiry date digits recognition using deep learning," *IEEE National Aerospace & Electronics Conference (NAECON 2019)*, pp. 302-304, Dayton, OH, USA, 2019.
- [8] Expiry Date Digits Dataset, [Online]. Available: <https://www.kaggle.com/tareqkhan/expiry-date-digits>, 2021.
- [9] LeCun, Y., Bengio, Y., and Hinton, G., Deep learning. *Nature*, 521(7553), pp.436-444, 2015.

- [10] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted Boltzmann machines." *In Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807-814. 2010.
- [11] Nagi, J., F. Ducatelle, G. A. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, L. M. Gambardella. "Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition", *IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011)*, 2011.
- [12] Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research*. Vol. 15, pp. 1929-1958, 2014.
- [13] Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." *In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249-256. 2010.
- [14] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification." *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026-1034. 2015.
- [15] Softmax activation function. Available online: <https://keras.io/activations/#softmax>, 2021.
- [16] R. Gandhi, A Look at Gradient Descent and RMSprop Optimizers, [Online]. Available: <https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizers-f77d483ef08b>, 2021.
- [17] Keras: The Python Deep Learning library, [Online]. Available: <https://keras.io/>, 2021.

Tareq Khan was born in Irvine, the UK in 1980. He received a B.Sc. degree from the Department of Electrical and Electronics Engineering of the Islamic University of Technology, Dhaka, Bangladesh, in 2003, and an M.Sc. degree from the Department of Electrical and Computer Engineering of Concordia University, Montreal, Canada, in 2007. He received a Ph.D. degree from the Department of Electrical and Computer Engineering of the University of Saskatchewan, Canada in 2014.

He is currently working as an Assistant Professor in the School of Engineering of Eastern Michigan University (EMU) in the USA. At EMU, he has received several research awards such as Summer Research Award, Faculty Research Fellowship, Game-Above Faculty First Funding, and James H. Brickley awards. To date, Dr. Khan has authored 60 publications in the area of Smart Home, Internet of Things (IoT), Machine Learning, Embedded Systems, and Image Processing. He currently has 2 US patents granted.

Dr. Khan is a member of the Institute of Electrical and Electronics Engineers (IEEE).