

RESEARCH

Open Access



CNN-optimized text recognition with binary embeddings for Arabic expiry date recognition

Mohamed Lotfy^{1,3} and Ghada Soliman^{2,3*}

*Correspondence:
Ghada.Soliman@orange.com

¹ Department Software
Engineering, Kafr El-Sheikh
University, Kafr El Sheikh, Egypt

² PhD, Department
Environmental Engineering, Ain
Shams University, Cairo, Egypt

³ Orange Innovation Egypt, Giza,
Egypt

Abstract

Recognizing Arabic dot-matrix digits is a challenging problem due to the unique characteristics of dot-matrix fonts, such as irregular dot spacing and varying dot sizes. This paper presents an approach for recognizing Arabic digits printed in dot matrix format. The proposed model is based on convolutional neural networks (CNN) that take the dot matrix as input and generate embeddings that are rounded to generate binary representations of the digits. The binary embeddings are then used to perform Optical Character Recognition (OCR) on the date images. To overcome the challenge of the limited availability of dotted Arabic expiration date images, we developed a True Type Font (TTF) for generating synthetic images of Arabic dot-matrix characters. The model was trained on a synthetic dataset of 3287 images and 658 synthetic images for testing, representing realistic expiration dates from 2019 to 2027 in the format of *yyyy/mm/dd* and *yy/mm/dd*. Our model achieved an accuracy of 98.94% on the expiry date recognition with Arabic dot matrix format using fewer parameters and less computational resources than traditional CNN-based models. By investigating and presenting our findings comprehensively, we aim to contribute substantially to the field of OCR and pave the way for advancements in Arabic dot-matrix character recognition. Our proposed approach is not limited to Arabic dot matrix digit recognition but can be also extended to text recognition tasks, such as text classification and sentiment analysis.

Introduction

Tracking of products' expiration dates is one of the crucial tasks in the medicine and food industries where consumer health is affected by the accuracy and efficiency of the detection systems. Consuming Expired products or drugs could have severe consequences such as life-threatening illness. QR code and barcodes were one of the solutions for effective detection, but those methods still depend on the human factor for filling in the database which is less efficient in both time and accuracy.

Recognizing digits is a fundamental challenge in computer vision, with practical applications such as optical character recognition and automated document processing. While significant progress has been made in recognizing Latin digits, recognizing Arabic digits presents a unique challenge due to the complex nature of Arabic script. Additionally, the availability of training data for recognizing Arabic digits, particularly in the

Arabic dot True Type Font (TTF) matrix format, is limited. Existing datasets often lack diversity, particularly in terms of variations in writing style and quality.

The field of digit recognition, and more specifically expiration date recognition, has been extensively studied in recent years, with researchers exploring various aspects of the subject matter to gain deeper insights and address existing challenges. Recent studies have shown that neural networks exhibit promising performance in recognizing expiration dates.

One of the proposed solutions by Gong et al. [8] pipeline is to detect and recognize the expiration date for an automatic expiration date recognition system. Firstly, the expiration date is detected by extracting the region of interest (ROI) using a deep neural network. Following, Image preprocessing techniques with maximally stable extremal regions (MSER). Component Connected Analysis and Canny edge detection are applied to make a binarization of the extracted data region with characters being differentiated from the background, identification of the blobs representing different characters, and then extraction of the boundaries of the digits, respectively. Tesseract OCR is then employed to segment the digits. Finally, the extracted shapes of the digits are then classified by the nearest neighbor method. The pipeline runs on filled-in images with Latin digits and Color image formats (color/ grayscale).

Muresan et al. [18] developed a pipeline to detect and recognize expiration dates on water bottles. The pipeline first uses a camera to capture an image of the bottle in a controlled environment with no light reflections. The image is then segmented using Mask RCNN [9] to crop the bottle and extract the expiration date. The ROI is then preprocessed by resizing, converting to grayscale, applying morphological gradient operations, and thresholding using Otsu's Algorithm [16]. Closed contours of the expiration date are then detected. Characters are segmented by finding gaps between them and resolving connected digits. Dot-matrix characters are reconstructed using dilation of a 3×3 filter for 2 iterations with OpenCV to fill in missing parts. A modified LeNet-5 [14] convolutional neural network architecture is used to recognize the segmented digits. The pipeline runs on filled-in images with Latin digits in grayscale format.

Florea and Rebedea [5] present a comprehensive solution for detecting and recognizing expiration dates. Their approach involves utilizing the TextBoxes++ architecture [15] which is based on a deep neural network, to extract regions of interest that potentially contain expiration dates. Subsequently, a convolutional recurrent neural network (CRNN) is fine-tuned using the cropped regions of interest to detect and decode the digits from the expiration date. To further process the detected dates, the authors employ a combination of regular expressions and logical criteria, along with a library capable of parsing time and date in popular formats. The authors employ a dataset comprising both real images, such as SynthText [7] and ICDAR [21], where the expiration dates are printed on products, as well as synthetic images generated using downloadable dot matrix type characters with the PIL package and Unity3D graphics. These synthetic images are then blended into the uneven surface of the objects. The pipeline is designed to operate on dot-matrix type characters or filled-in images generated by thermal printers. The characters are in Latin and colored image format.

Ashino and Takeuchi [2] propose a pipeline that combines two deep neural networks to detect and recognize expiration dates on drink packages. The pipeline involves using

object detection to locate the region containing the expiration date and identify the characters (digits and delimiters) present. Subsequently, a character-recognition deep neural network (DNN) is utilized to recognize the characters after they have been extracted from the images. This pipeline is specifically designed to operate on Latin dot matrix characters and colored image format.

In Khan [11] study, a convolutional neural network (CNN) model is presented for recognizing the digits in expiration dates. The pixel data type is converted from integer to floating-point for improved performance. The author curated a dataset consisting of 1000 pictures, encompassing 10 types of digits ranging from 0 to 9. Each digit is represented by 100 images. The images are subsequently resized and cropped to a dimension of 32×32 pixels. The CNN model operates on single digit recognition that can be in dot-matrix format or filled-in images containing Latin digits in a colored image format.

In their updated work, Gong et al. [6] present a pipeline designed to detect and recognize expiration dates on food package images. The researchers employ a fully convolutional neural network to extract the expiration date information, followed using CRNN for digit recognition. The CNN model operates on images that contain Latin digits and are in colored format. Seker and Ahn [20] propose a three-step framework for detecting and recognizing expiration dates on product packages. They use the Fully Convolutional One-Stage (FCOS) model [23], originally designed for object detection, to detect and extract the expiration date region from input images. To detect the day, month, and year components within the extracted region, the authors adapt FCOS by removing the feature pyramid network (FPN) to reduce network complexity in the DMY detection network. For character recognition, they adapt the decoupled attention network (DAN), which was originally developed for scene and handwritten text recognition, to recognize the characters in the day, month, and year regions. To fine-tune the DAN model, which was primarily trained on scene and handwritten text images, the authors use a dataset of synthetic date images with various expiration date font types and 13 date formats. The framework is designed to work on colored image formats containing either filled-in or dot matrix characters in Latin.

Our paper proposes a novel approach for the recognition of Arabic dot-matrix characters using a lightweight CNN-based model that generates binary embeddings as shown in Fig. 1. These embeddings will be then rounded at a threshold of 60% to generate the binary representation of the digits. By reducing the number of classes in the final linear layer, our approach can decrease the model size in classification tasks without compromising accuracy. Hence, our model was able to attain accuracy by achieving an accuracy of 98.94% on the expiry date recognition with Arabic dot matrix format with fewer

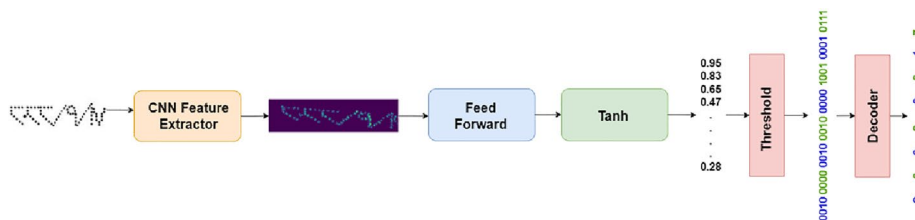


Fig. 1 Main Components of CNN-Optimized with Binary Embeddings model Architecture

parameters and less computational resources compared to conventional CNN-based models.

Our contributions in this study can be summarized into the following:

- As there is a lack of literature addressing the product expiry date in Arabic dot matrix, we propose a novel lightweight CNN with an integrated binary optimization technique designed for decoding Arabic dot matrix formatted expiration dates in the date format of *yyyy/mm/dd* and *yy/mm/dd*. This model can be easily customized to accommodate various input date formats.
- We evaluated our approach of incorporating binary embeddings into the model architecture using the VGG16 backbone [22] and MobileNetV2 [19] on the CIFAR-100 input dataset. The results indicated a noteworthy reduction in 119 K model parameters for both VGG16 and MobileNetV2. By applying the binary embedding technique, the model achieved classification of up to 127 classes using merely 7 neurons in the final layer, as opposed to the conventional 127 neurons.
- In the absence of an existing public dataset for Arabic dot matrix expiration dates, we developed a new dataset that comprises challenging images. These images feature synthetic expiration dates in the Arabic dot matrix format, deliberately introducing inconsistencies such as uneven spacing and misalignment, which may potentially compromise readability.

The paper is organized into the following: section "[Dataset Generation](#)" describes the challenges associated to the dot matrix format and the approach taken to generate synthetic images. In section "[Methodology](#)", the overall methodology including the main components of the architecture are explained. Followed by section "[Results](#)" that introduces the results of our work with comparing it to the previous work in the literature, the comparative analysis between sigmoid and tanh activation functions, and the study of the pretrained models. Finally, section "[Conclusion](#)" includes the conclusion of our work.

Dataset generation

Generating a dataset for dot matrix expiration dates poses unique challenges due to the absence of readily available datasets for Arabic dot matrix formats, necessitating the creation of a novel dataset. The process involves introducing synthetic expiration dates with deliberate complexities, such as uneven spacing and misalignment. These intentional inconsistencies are essential to simulate real-world scenarios where the dot matrix representation may encounter variations in printing quality or display in terms of uneven spacing, low-resolution appearance, and rotated dot-matrix characters. Additionally, the dataset generation process must carefully balance the creation of challenging images without compromising the overall readability of the expiration dates, ensuring that the dataset accurately reflects the potential difficulties faced by recognition models in handling dot matrix formats. Addressing these challenges is crucial for training robust and effective models capable of accurately interpreting and extracting information from


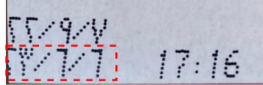
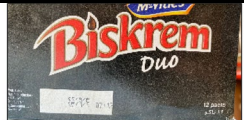
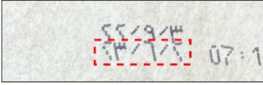
Products	Production Expiry Date	Expiry Date
		6/6/2023
		2/6/2023

Fig. 2 Examples of Natural Products with Expiry Dates in Arabic

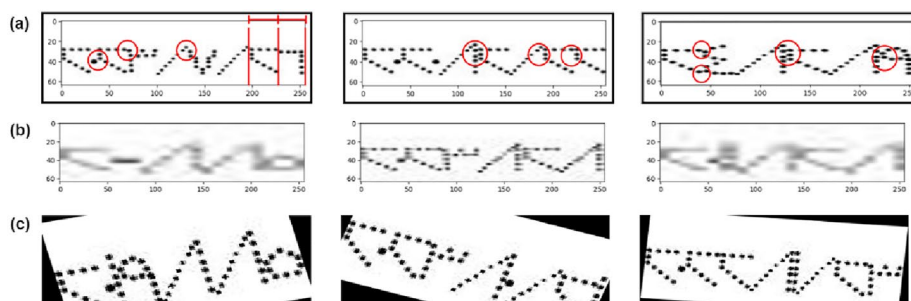


Fig. 3 Examples of Synthetic Dataset with Challenging Scenarios generated using PIL package in Cairo Font Style. **a** Uneven Spacing, **b** Low-Resolution Appearance, **c** Rotated Images

Arabic dot matrix expiration dates in practical applications. Figure 2 shows examples of natural products with expiry dates in Arabic.

Challenges

Lack of real data The absence of real data presents a notable challenge in the Arabic-speaking world when it comes to expiry dates on food and medical products. Arabic expiry date can be expressed in various formats leading to a lack of standardization in the format, making it difficult for consumers and retailers to accurately recognize the expiry dates. Unfortunately, there is currently no publicly available dataset for Arabic dot matrix format with various formats and scales to address this issue.

Traditional filling methods Conventional erosion and dilation techniques have been extensively employed for filling in dotted digits in various languages to facilitate digit recognition tasks. However, in the case of Arabic dotted digits, these techniques have demonstrated ineffectiveness for our custom synthetic dataset. The main reason behind this limitation is the almost negligible spacing between Arabic digits in our dataset, which poses a challenge for traditional erosion and dilation methods to precisely reconstruct the dots, as depicted in Fig. 3a.

Low-resolution appearance The low-resolution appearance of dot matrix characters can be attributed to various factors. The factor of Low Dot Density indicates that a fewer dots per inch resulted in less ink on the paper, leading to lighter and less vibrant images. In addition, the ribbon that has lost its ink or has uneven ink distribution will produce faded or inconsistent images. Examples of these synthetic images with low-resolution appearance are shown in Fig. 3b.

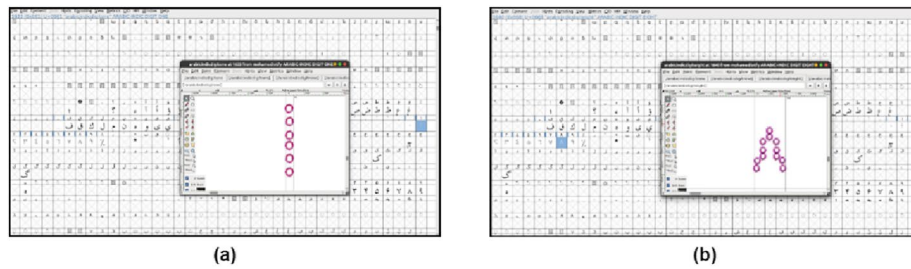


Fig. 4 Drawing the TrueType Font (TTF) on FontForge. **a** Arabic dot matrix format corresponding to Latin digit (one), **b** Arabic dot matrix format corresponding to Latin digit (eight)

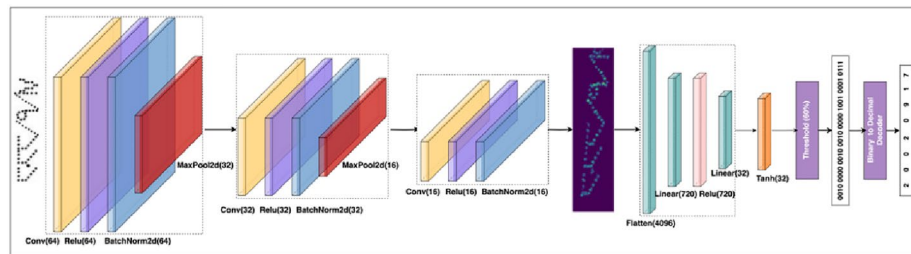


Fig. 5 CNN-Optimized with Binary Embedding for Arabic Expiry Date Recognition Architecture

Rotated images The fonts are appropriately designed for displaying the dot matrix characters in an upright form. Nevertheless, in certain newer display applications, such as a moving map display or a CAD/CAM system, the dot-matrix patterns might be rotated due to change in the relative position of the dots resulting in distortion to the character pattern [13]. The synthetic images were randomly rotated from 0 to 10 degrees for the dataset generation. Figure 3c shows examples of the rotated images from the synthetic dataset images.

Generate synthetic data using Arabic dot-matrix TTF

The synthetic dataset is created using Arabic dot-matrix TrueType Fonts (TTF), where the characters are initially drawn as vector graphics and subsequently saved as TrueType Font format through the utilization of FontForge, as illustrated in Fig. 4.

To generate synthetic Arabic dot-matrix characters, we employed the PIL package and used the created Arabic True Type Font that includes digits 0–9 with varying widths but consistent height, along with a delimiter symbol (“/”). Incorporating different digit widths in the font adds a distinctive visual aspect to the design and enhances model generalization by increasing the variability of input dot-matrix images.

Methodology

We introduce a novel technique for the recognition of the expiration digits, trained on the synthetic images of Arabic dot matrix format. Figure 5 shows the main layers of CNN-based model with Binary Embeddings, used for Arabic Expiry Date Recognition.

The purpose of our work is focused on developing a lightweight and high-speed model by generating embeddings in the form of binary representations, that will be then decoded into expiry dates. Our model produces an output of 36 probabilities

representing how likely each bit should be enabled as 1 or disabled as 0 when applying a threshold of 60%. The 0.5 threshold is commonly set as a default for machine learning, however it might not be optimal for real-world applications, so we choose a range of 10% to avoid any randomness of the result and the best result obtained was of applying a threshold of 60% on the Tanh output. For example, the date format of yyyy/mm/dd holds a fixed sequence of 8 digits where each digit is encoded into a 4-bit binary representation; for example, a digit of 9 is encoded into a binary representation of 1001. It can be also trained on images with different date formats while maintaining the same binary representation output, as for example January will be encoded into the binary representation of 01 and so forth.

Algorithm 1 Generate Binary Embeddings for Expiry Date Recognition Algorithm

Require: Image I

Ensure: Binary Vector B

1: $D \leftarrow \text{ExtractDateLabel}(I)$

2: $B \leftarrow \text{ConvertToBinary}(D)$

3: $X_{train}, X_{test}, y_{train}, y_{test} \leftarrow \text{SplitData}(B, 0.8)$

4: $model \leftarrow \text{TrainBinaryModel}(X_{train}, y_{train})$

5: $y_{pred} \leftarrow \text{Predict}(model, X_{test})$

6: $B_{pred} \leftarrow \text{RoundToThreshold}(y_{pred}, 0.6)$

7: **Algorithm Description:**

8: This algorithm takes an input image I and outputs a binary vector B that represents the classification of the image. The algorithm works by first extracting the date label from the image and converting it to a binary representation. The dataset is then split into training and testing sets, and a binary classification model is trained using the training set. The model is then used to predict the binary label of the test set. Finally, the predicted labels are rounded using a threshold of 0.6 to obtain the final binary vector B_{pred} .

9: **return** B_{pred}

The traditional models aim to activate the neuron corresponding to the target class. In the proposed method, binary representation requires the model to activate multiple neurons simultaneously. For instance, determining class 9 involves firing the neurons with the binary representation 0001011. Our approach of using the binary embedding layer employs two characteristics. The first one relies on the size reduction where the number of neurons equals the number of classes in traditional classification layers, posing challenges with large numbers of classes. The approach converts the total number of classes to its binary representation, significantly reducing the required number of neurons. For example, a 100-class problem would only need 7 neurons in binary representation, leading to a more compact model. Secondly, increasing generalization as of instead of firing a single neuron for a specific class, the model must activate a combination of neurons in the classification layer. This leads to enhanced generalization as the model explores various combinations of neurons in the preceding layer before the classifier to make accurate predictions.

CNN feature extractor

The initial stage of our model involves the extraction of dates within an image, employing Convolutional Neural Networks (CNNs) due to their effectiveness in capturing key features, particularly the spatial location of the date within the image [24]. Our model incorporates three convolutional blocks, each of the first two blocks is comprised of CNN Layer, rectified linear unit (ReLU) activation function, batch normalization, and the Max Pooling Layer. The final convolutional block consists of CNN Layer, rectified linear unit (ReLU) activation function, and batch normalization, without the Max Pooling Layer. The input image is of dimension $256 \times 64 \times 1$. The output of the final convolutional block represents the CNN Feature extractor that is of size $64 \times 16 \times 4$, which is subsequently flattened as an input into the dense layer.

Equation 1 shows the output a_{ij} calculated after applying convolution operation at the next layer of location (i, j) [17]. The maximum pooling retains the most distinguished features of the input image by reducing its dimensions. The mathematical formula of max pooling is given in Eq. 2. Batch Normalization is applied to each dimension independently by ensuring that each feature has a zero mean and unit variance [10] as seen in Eq. 3. It also provides regularization to the model with diminishing the need for Dropout.

$$a_{ij} = \sigma((W * X)_{ij} + b) \quad (1)$$

where X is the input provided to the layer, W is filter or kernel which slides over input, b is the bias, $*$ representing the convolution operation, and σ is nonlinearity introduced in the network.

$$O(i, j) = \max(I(i + m, j + n)) \quad (2)$$

where I and j are the indices of the output pooled feature map, m and n iterate over all positions within the filter size, and \max calculates the maximum value of the feature map.

$$y = \frac{x - E[x]}{\sqrt{\text{Var}[x] + \varepsilon}} * \gamma + \beta \quad (3)$$

where y is the output of the batch normalization layer, x is the input to the batch normalization layer, $E[x]$ is the mean of the activations across the batch for each feature dimension, $\text{Var}[x]$ is the variance of the activations across the batch for each feature dimension, γ (gamma) is a learnable scale parameter, β is a learnable shift parameter, and ε is a small constant value added to the variance for numerical stability.

CNNs possess remarkable capabilities in extracting intricate patterns and establishing relationships within image data. They excel at binary prediction tasks by autonomously discerning between different image classes based on the features they learn from the data, without necessitating manual feature engineering or domain-specific knowledge.

Our model utilizes three convolutional layers with respective channel numbers of 64, 32, and 16, each accompanied by a rectified linear unit (ReLU) activation function. Each layer is responsible for extracting distinctive features within the date image. The max

pooling operations are also employed to reduce dimensionality and downsample the data, thereby diminishing variability. We employ two max-pooling layers with a kernel size of 2 for all layers, except for the final layer. The batch normalization normalizes the activations across the batch dimension, and is commonly used in convolutional neural networks (CNNs) to stabilize and accelerate the training of convolutional layers.

Feed forward linear layer

The output of the CNN layer is flattened to downsample the image into a dense vector containing all the features of the convolution channels. In our model, two linear layers were used; the first dense layer outputs 720 neurons while the second dense layer outputs 32 neurons, representing the number of binary classes. The expiration date is comprised of 8 characters where each character is represented by 4 neurons (i.e., 4 bits).

Activation function

The activation function is crucial as different activation functions may lead to different accuracy. In our work, we conducted experiments using sigmoid and tanh activation functions to produce probabilities ranging from 0 to 1.

Sigmoid was a convenient choice as it produces probabilities between 0 and 1. However, in our case, tanh outperforms sigmoid due to its wider range and faster convergence [4]. The characteristic of the sigmoid function tends to push the input values to either end of the curve (0 or 1) due to its S-like shape. On the other hand, the tanh function is considered as a stretched and shifted version of sigmoid. Due to the positive nature of the features, the tanh activation function did not produce any negative activations and hence the thresholding provides with better results on the tanh outputs over the sigmoid ones. As can be shown in Fig. 6, tanh function is symmetric around the origin where the activation will be positive if the input is positive. As a result, the curve will be shifted to the right, and the output values tend to be in the range of between 0 and 1.

Generate binary embedding for Arabic expiry date recognition

The probabilities output from the Tanh function, our choice of the activation function, are then rounded into 1's and 0's using a threshold of 60%. As such, this binary

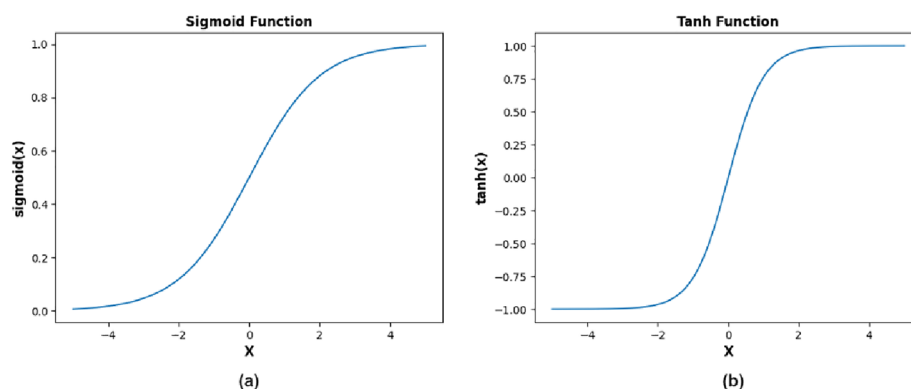


Fig. 6 Comparison between the function of Sigmoid and Tanh. **a** Sigmoid Function with S-like shaped curve, **b** Tanh Function with stretched and shifted version curve

representation can be viewed as an embedding that will be utilized for the expiry date recognition.

Loss function

To ensure the accurate training of our model, it is crucial to carefully consider the choice of the loss function. Initially, we experimented with mean squared error (MSE) and Binary Cross-Entropy (BCE) as they are commonly used for tasks involving input–output approximation.

The learning process of the model was greatly improved when we employed Binary Cross Entropy with logits loss (BCEWithLogitsLoss) that is supported by Pytorch, a unified class that combines a sigmoid layer with BCELoss. This approach proves to be more numerically stable due to the advantage of the logsum-exp trick compared to using a standalone Sigmoid followed by BCELoss.

The log-sum-exp trick is a numerical technique, used to improve the stability of calculations involving exponential functions. It is often used in machine learning and other fields that involve large numbers or probabilities. It involves taking the logarithm of the sum of exponentials, rather than calculating the sum of exponentials directly. This can help to avoid numerical overflow or underflow, which can occur when working with very large or very small numbers.

Results

The proposed model results

This section presents the performance of our model for decoding the Arabic dot-matrix expiry dates images using the activation functions of sigmoid and tanh as shown in Table 1. The model was trained on a synthetic dataset of 3287 images and 658 synthetic images for testing, representing realistic expiry dates with challenging cases from 2019 to 2027 in the format of *yyyy/mm/dd* and *yy/mm/dd*. Our model achieves a high accuracy of 98.94% using tanh compared to an accuracy of 95.59% using sigmoid. The test accuracy is computed as the ratio between the number of correct predictions of the decoded expiry dates within the test images to the total number of the test images with referring to Eq. 4 where any detection of wrongly decoded character of the date would be considered as misclassified of the whole test image. Our model runs on GPU specification of Nvidia GTX 1050 Ti for both training and inference and was able to decode 1 k images per second at inference.

$$\text{Test Accuracy} = \frac{\text{Number of Correct Predictions of Decoded Expiry Dates}}{\text{Total Number of Images}} \tag{4}$$

Table 1 Comparison of our model performance using Sigmoid and Tanh

Input size	No. of samples train/ test	Inference Speed (img/s)	Convergence @epoch	Activation function	Test accuracy (%)
256 × 64 × 3	2629/658	1 k	50	Sigmoid	95.59
256 × 64 × 3	2629/658	1 k	50	Tanh	98.94

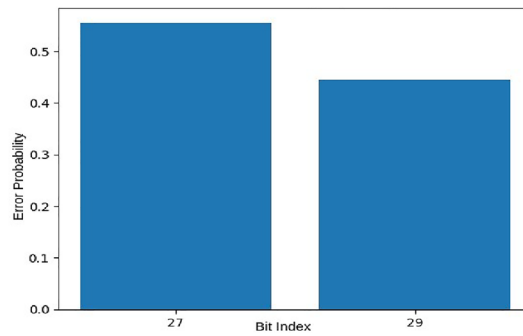


Fig. 7 Error occurs only at Bits Index of 27 and 29 in the output Vector

Table 2 Summary of Approaches on the Expiry Date Recognition Methods

Approach	Dataset (Synthetic/ Real)	Image Format	Image Content	Latin/ Arabic	Filled in/ Dot-matrix	Inference Speed (img/s)	Recognition Approach	Test Accuracy (%)
Florea and Rebedea [5]	Both	Colored	Expiry Date	Latin	Both	0.25	RCNN	72.7
Ashino and Takeuchi [2]	Real	Binarized	Single Digit	Latin	Dot-matrix	–	DNN	90
Khan [11]	Real	Colored	Single Digit	Latin	Both	0.89	CNN	90
Gong et al. [6]	Real	Colored	Expiry Date	Latin	Dot-matrix	–	CRNN	95
Seker and Ahn [20]	Both	Colored	Expiry Date	Latin	Both	0.12	DAN	97.74
Ours	Synthetic	Colored	Expiry Date	Arabic	Dot-matrix	1 k	CNN with Binary Embedding	98.94

The highlighted information in bold demonstrates the significance of our work on the expiry date recognition in Arabic dot-matrix format.

We also investigated the failed examples on the results obtained with tanh. As shown in Fig. 7, we found that neurons 27 and 29 were the only neurons generating errors, specifically related to the days of the date. This error occurrence can be attributed to the fact that these neurons are not activated by any number except for nine and eight. These digits contribute to low occurrence of the remaining digits during the month where they only enabled within 20% of the neurons, leading to suboptimal weights and limited propagation within these neurons.

A comparison of our work against the recent previous works on the expiry date recognition models is illustrated in Table 2. It can be noticed that most prior approaches predominantly focus on the expiry date recognition, whereas Ashino and Takeuchi [2] and Khan [11] addresses the digit recognition. The authors utilized datasets containing Latin characters, while our approach involved synthetic dataset with Arabic characters. Recognizing Arabic digits exhibits numerous challenges compared to Latin digits, including variations in writing style, size, shape, and slant, along with

the presence of image noise, all contributing to alterations in numeral topology [1], and hence this variability can lead to increased ambiguity during the recognition process of Arabic digits. Our approach surpasses the authors' work in terms of inference speed, demonstrating superior efficiency in processing time in which our model can process a rate of 1 k images per second during the inference time compared to the inference speed time taken by the approaches of Florea and Rebedea [5], Khan [11], and Seker and Ahn [20] as shown in Table 2. Our approaches underscore the significance of our work on developing a lightweight and high speed model for the recognition of the expiry date in Arabic dot matrix format.

Comparative analysis between sigmoid and tanh activation functions

We studied the impact of using both the sigmoid and tanh activation functions on the probabilities after being rounded to 0 and 1 on the testing dataset. As shown in Fig. 8, the binary representation of 3 at the position of the day, was not obtained correctly when thresholding on the probabilities output of the sigmoid function. Throughout this analysis, it was observed that the hyperbolic sigmoid was not reliable as the curve tends to be smoother in the middle when the values become closer to 0.5. Hence, it makes it difficult to have a clear separation among the probabilities after thresholding to generate the proper binary representation. On the other hand, tanh has proved to be more reliable in our domain context of expiry date recognition due to noticeable variations in the probabilities hence impacting the binarization process to generate accurately the binary representation of the digits.

Study of binary embedding approach on pretrained models

In order to assess the capabilities of our approach, we extended our experiments to include two pretrained models, VGG16 [22] and MobileNetV2 [19], and evaluated them on different domain context using the CIFAR100 dataset [12] that is different from the expiry dates recognition use case. CIFAR100 dataset consists of 60,000 samples with 100 classes with 600 samples each.

The pretrained model weights of the backbone of VGG16 and MobileNetV2 were frozen that were mainly trained on ImageNet images, a large visual database designed for use in visual object recognition software research which consists of over 15 million labeled high-resolution images in over 22,000 categories [3]. Two experiments were conducted on each of VGG16 and MobileNetV2 backbones by adding a dense layer of 100 neurons and 7 neurons in separate followed by Tanh activation function as shown in Fig. 9. With using the dense layer of 7 neurons, it was trained on CIFAR-100 dataset. The neurons of this dense layer represent the binary embedding of any of the 100 classes. For example, the binary representation of 1100011 is decoded to class 99 by thresholding the probabilities output of the 7 neurons without the need of having 100 neurons mapped to 100 classes.

By the approach of using the binary embedding technique, the binary representation of 1111111 will accommodate up to 127 classes while maintaining the same model size and the number of neurons at the final layer. Moreover, adding an extra neuron to the

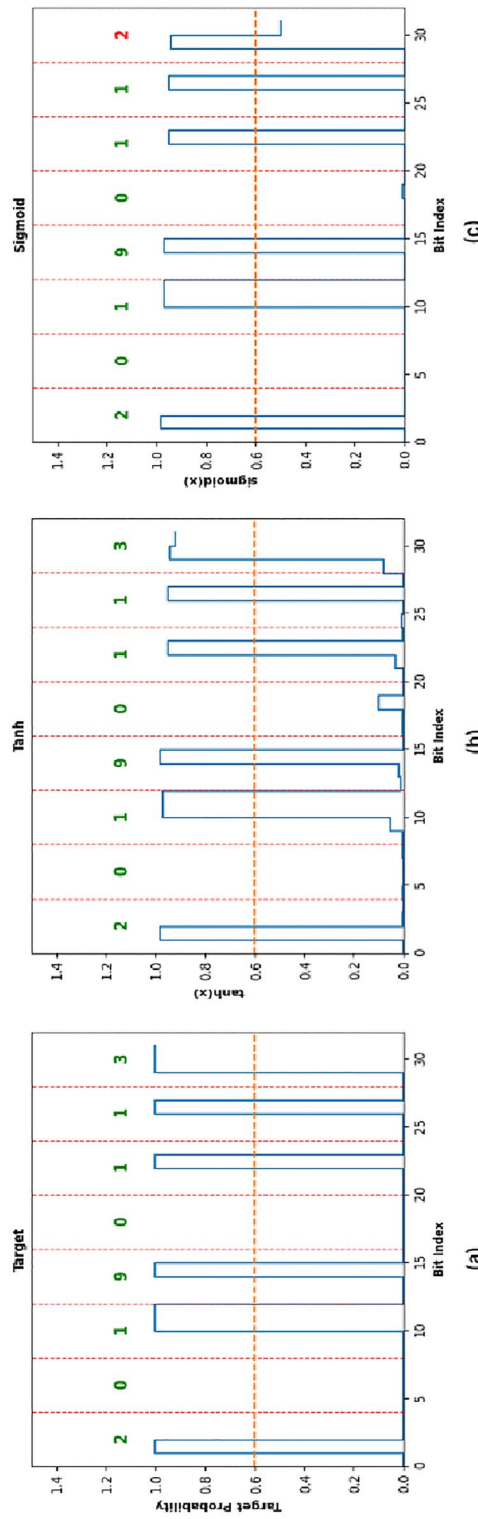


Fig. 8 Comparative Analysis between the activation functions of Sigmoid and Tanh without applying the Thresholding against the target embeddings. **a** Target Binary Representation of Expiry Date, **b** Actual Model Outputs with Probabilities of Tanh with Mapping them to the corresponding characters of the expiry date when the thresholding is to be applied, **c** Actual Model Outputs with Probabilities of Sigmoid with Mapping them to the corresponding characters of the expiry date when the thresholding is to be applied. Orange color indicates the threshold of 60%. Red color indicates the wrong character to be obtained after Decoding

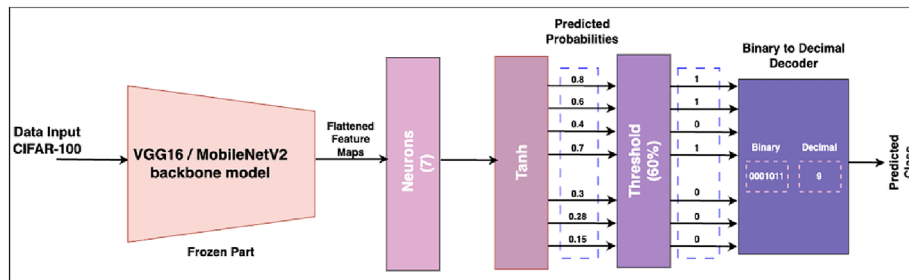


Fig. 9 The model backbone architecture (VGG16/ MobileNetV2) with dense layer of 7 neurons with thresholding the prediction probabilities by 60% followed by the Binary to Decimal Decoder for decoding the binary representation into the predicted class

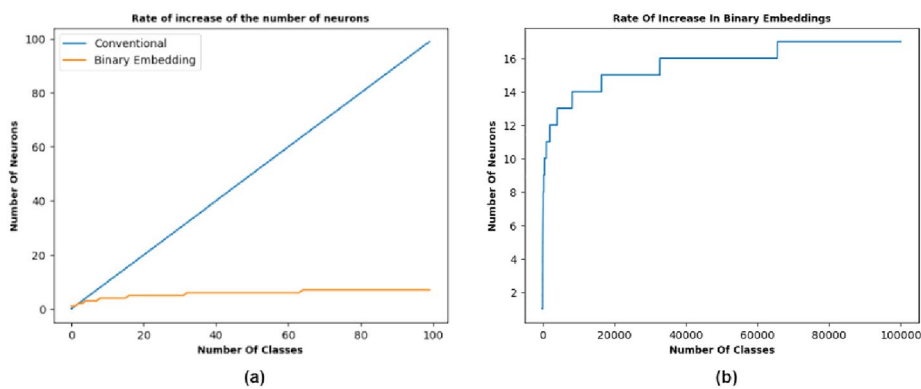


Fig. 10 a The rate of change between the number classes and number of neurons using conventional and binary embedding, **b** The rate of increase of the number of neurons and its impact on the number of classes using the Binary Embedding technique

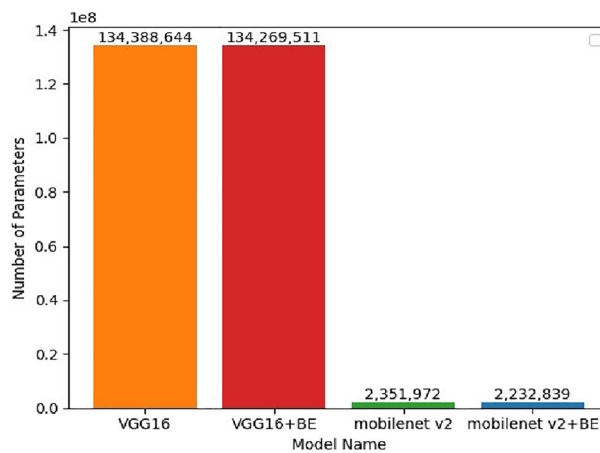


Fig. 11 Model sizes with and without Binary Embedding (BE) versus the Number of Parameters

final layer will increase the model capacity to accommodate 127 extra classes with a total of 254 classes.

Figure 10 illustrates the linear relationship between the number of neurons corresponding to the number of classes while it almost appeared as a steady line for the same

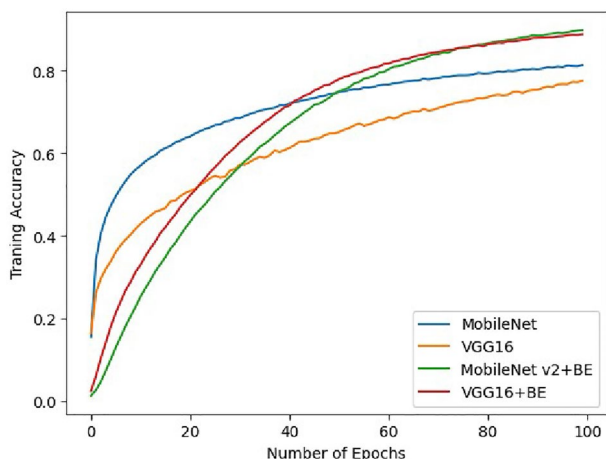


Fig. 12 Comparison of the accuracy against the number of epochs during training among the models: MobileNetV2 baseline, VGG16 Baseline, MobileNetV2 + Binary Embedding (BE), and VGG16 + Binary Embedding (BE)

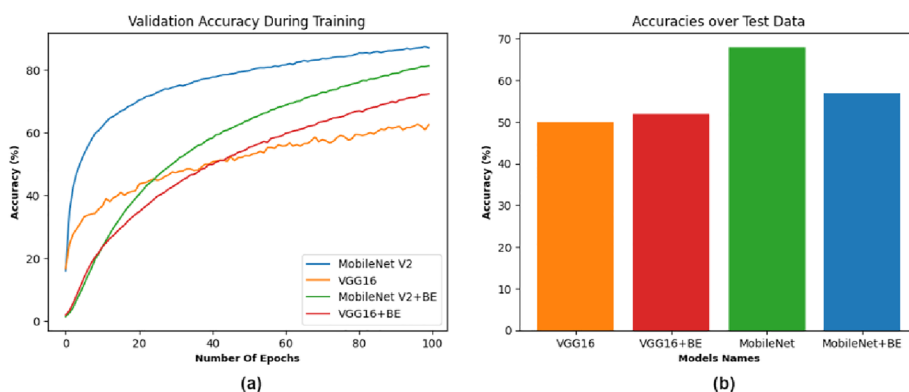


Fig. 13 Comparison of the validation and testing accuracy against the number of epochs during training among the models: MobileNetV2 baseline, VGG16 Baseline, MobileNetV2 + Binary Embedding (BE), and VGG16 + Binary Embedding (BE). **a** Number of Epochs versus Validation Accuracy, **b** Test Accuracy of used Models

relationship, demonstrating insignificant difference on the number of neurons when the number of classes reached up to 100 classes. It also demonstrates a slower rate of change that indicates a slower rate of growth on the number of neurons when the number of classes is increased.

Additionally, we observed that the model parameters are reduced by 119 K for both VGG16 and MobileNetV2 as shown in Fig. 11 due to leveraging the binary embedding in the last classification layer.

The models of VGG16 and MobileNetV2 were trained for 100 epochs, batch size of 64, and Adam optimizer with 0.9 momentum. The default data augmentation of ImageNet has been primarily used by applying the mean normalization of RGB values of (0.485, 0.456, 0.406) and standard deviation normalization of values of (0.229, 0.224, and 0.225) for each of VGG16 and MobileNetV2. Figure 12 illustrates that the accuracy curve of the models with binary embedding started below the traditional

models but eventually surpassed their performance. We observed overfitting during the training, deducing an increase in the model complexity with a reduction in model parameters. To overcome this overfitting, we applied augmentation techniques to increase the dataset, such as horizontal flip, rotation, zoom, shear, and color jitter.

As shown in Fig. 13 the binary shows competitive results against the convention pre-trained models on the validation and testing accuracy. According to the obtained results on the testing accuracy, VGG16 with Binary Embedding exceeds the base model of VGG16 by 2% while obtaining an accuracy decrease of 11% less than that of the base model of MobileNetV2. We noticed that VGG16 and MobileNetV2 models with binary embedding did not reach a saturation state at 100 epochs as the smoothness of the validation curve indicates a continuous increase in the validation accuracy of these models.

We also examined the binary embedding of 10 classes generated from VGG16 model with Binary output layer. It is shown from Fig. 14 that the model was able to successfully generate the embeddings of these classes resulted in cohesive clusters and therefore effectively distinguished among classes.

Moreover, the optimized models utilizing the binary technique reduced the computational power and time costs. The detailed analysis showed that our novel technique effectively reduced model size and computational complexity while maintaining high accuracy compared to the traditional technique. We are optimistic that further optimization of this technique could open a wide range of applications for its deployment.

Conclusion

In conclusion, we have successfully developed a comprehensive model for text recognition using a realistic dataset comprising dates. Our model architecture is based on a lightweight CNN with an integrated binary optimization technique. Our research findings indicate that the most favorable results are achieved when employing an architecture that incorporates the tanh activation function in conjunction with binary cross-entropy with logits, and convergence after 50 epochs.

By leveraging binary embedding at the output layer, our model was able to achieve an impressive accuracy of 98.94% on decoding the Arabic expiration dates on the test dataset. Importantly, we considered any date with a single erroneous character as

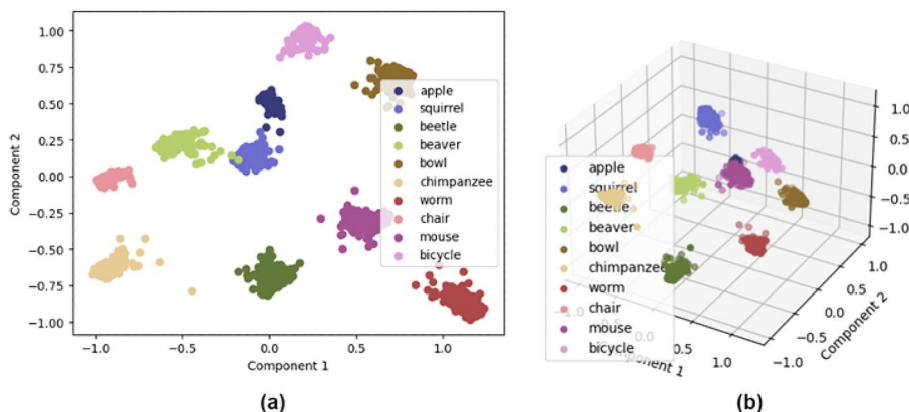


Fig. 14 Clusters of Binary Embedding of a subset of 10 classes from CIFAR-100 with VGG16+BE. **a** Visualization of Embeddings Clusters in 2 dimensions, **b** Visualization of Embeddings Clusters in 3 dimensions

misclassified. The importance of decoding Arabic dot-matrix digits cannot be overstated, given the scarcity of research or papers addressing this specific aspect of optical character recognition (OCR). Our focus on decoding Arabic dot-matrix digits assumes great significance due to the absence of relevant literature on decoding Arabic dot-matrix format. This highlights the necessity to explore and develop robust methodologies to effectively tackle this challenging problem. By presenting our comprehensive findings, we aim to make a substantial contribution to the field of OCR and pave the way for advancements in recognizing Arabic dot-matrix digits.

Our work presents an area of development for future exploration that lies in the promising potential of binary embedding on reducing the model size without compromising the performance of the model. Evaluating our model efficiency and assessing its performance on other pretrained models (VGG16 and MobileNetV2) by using alternative datasets as for CIFAR100 from various domains demonstrate our model generalizability and applicability to other domain contexts with achieving competitive performance over other pretrained models.

In summary, our results demonstrate the effectiveness of our text recognition model in handling realistic dates, thanks to the novel task-agnostic optimization technique employed. Moreover, our findings indicate the potential application of our approach to a wide range of other classification problems that represent a significant avenue for future research.

Abbreviations

BCE	Binary cross-entropy
BE	Binary embedding
CAD	Computer-aided design
CAM	Computer-aided manufacturing
CNN	Convolutional neural network
CRNN	Convolutional recurrent neural network
DAN	Deep averaging network
DNN	Deep neural network
FCOS	Fully convolutional one-stage
FPN	Feature pyramid network
GPU	Graphics processing unit
MSE	Mean squared error
MSER	Maximally stable extremal regions
OCR	Optical character recognition
QR	Quick-response
RCNN	Region-based convolutional neural network
ReLU	Rectified linear unit
ROI	Region of interest
TTF	True type font

Acknowledgements

Not applicable.

Author contributions

All authors contributed to the study conception and design. All authors have participated in material preparation, data collection and analysis. All authors read and approved the final manuscript.

Funding

This research received no external funding.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 15 November 2023 Accepted: 23 January 2024

Published online: 21 February 2024

References

- Alani AA (2017) Arabic handwritten digit recognition based on restricted boltzmann machine and convolutional neural networks. *Information* 8:142
- Ashino M, Takeuchi Y (2020) Expiry-date recognition system using combination of deep neural networks for visually impaired. In: *Computers helping people with special needs: 17th international conference, ICCHP 2020, Lecco, Italy, September 9–11, 2020, proceedings, Part I* 17. Springer, pp 510–516
- Deng J, Dong W, Socher R, Li-Jia L, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp 248–25. <https://doi.org/10.1109/CVPR.2009.5206848>.
- Dubey S, Singh S, Chaudhuri B (2022) Activation functions in deep learning: a comprehensive survey and benchmark
- Florea V, Rebedea T (2020) Expiry date recognition using deep neural networks. *Int J User-Syst Interaction* 13(1):1–17
- Gong L, Thota M, Yu M, Duan W, Swainson M, Ye X, Kollias S (2021) A novel unified deep neural networks methodology for use by date recognition in retail food package image. *SIVIP* 15(3):449–457
- Gupta A, Vedaldi A, Zisserman A (2016) Synthetic data for text localisation in natural images. *CVPR* 1:3
- Gong L, Yu M, Duan W, Ye X, Gudmundsson K, Swainson M (2018) A novel camera based approach for automatic expiry date detection and recognition on food packages. In: *Artificial intelligence applications and innovations: 14th IFIP WG 12.5 international conference, AIAI 2018, Rhodes, Greece, May 25–27, 2018, proceedings* 14. Springer, pp 133–142
- He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*, pp 2961–2969
- Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *ICML'15: proceedings of the 32nd international conference on international conference on machine learning*, pp 448–456
- Khan T (2021) Expiry date digit recognition using convolutional neural network. *Eur J Electr Eng Comput Sci* 5(1):85–88
- Krizhevsky A (2009) Learning multiple layers of features from tiny images
- Kurokawa K, Decker JJ, Kelly PL, Snyder HL (1988) The effects of image rotation on dot-matrix characters. *Proc Hum Factors Soc Ann Meet* 32(19):1391–1394
- LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
- Liao M, Shi B, Bai X (2018) A single-shot oriented scene text detector. *arXiv preprint arXiv:1801.02765*
- Liu D, Yu J (2009) Otsu method and k-means. In: *9th International conference on hybrid intelligent systems*, vol 1. IEEE, 344–349
- Maggiori E, Tarabalka Y, Charpiat G, Alliez P (2016) Fully convolutional neural networks for remote sensing image classification. In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp 5071–5074. <https://doi.org/10.1109/IGARSS.2016.7730322>
- Muresan M, Szabo P, Nedeveschi S (2019) Dot matrix ocr for bottle validity inspection. In: *2019 IEEE 15th international conference on intelligent computer communication and processing (ICCP)*. IEEE, pp 395–401
- Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4510–4520
- Seker A, Ahn S (2022) A generalized framework for recognition of expiration dates on product packages using fully convolutional networks. *Expert Syst Appl* 203:117310
- Shahab A, Shafait F, Dengel A (2011) Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In: *2011 international conference on document analysis and recognition*. IEEE, pp 1491–1496
- Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
- Tian Z, Shen C, Chen H, He T (2019) Fcos: Fully convolutional one-stage object detection. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 9627–9636
- Yamashita R, Nishio M, Do R, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9:611–629. <https://doi.org/10.1007/s13244-018-0639-9>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.