

# Machine learning based augmented reality for improved learning application through object detection algorithms

Anasse Hanafi, Lotfi Elaachak, Mohammed Bouhorma

Computer Science, Systems and Telecommunication Laboratory, Faculty of Sciences and Technologies, University Abdelmalek Essaadi, Tangier, Morocco

---

## Article Info

### Article history:

Received Mei 27, 2022

Revised Sep 16, 2022

Accepted Oct 13, 2022

---

### Keywords:

Computer vision

Detection transformer

Neural network

Object detection

Transformer

---

## ABSTRACT

Detection of objects and their location in an image are important elements of current research in computer vision. In May 2020, Meta released its state-of-the-art object-detection model based on a transformer architecture called detection transformer (DETR). There are several object-detection models such as region-based convolutional neural network (R-CNN), you only look once (YOLO) and single shot detectors (SSD), but none have used a transformer to accomplish this task. These models mentioned earlier, use all sorts of hyperparameters and layers. However, the advantages of using a transformer pattern make the architecture simple and easy to implement. In this paper, we determine the name of a chemical experiment through two steps: firstly, by building a DETR model, trained on a customized dataset, and then integrate it into an augmented reality mobile application. By detecting the objects used during the realization of an experiment, we can predict the name of the experiment using a multi-class classification approach. The combination of various computer vision techniques with augmented reality is indeed promising and offers a better user experience.

*This is an open access article under the [CC BY-SA](#) license.*



---

## Corresponding Author:

Anasse Hanafi

Computer Science, Systems and Telecommunication Laboratory, Faculty of Science and Technologies,

Abdelmalek Essaadi University

Tangier, Morocco

Email: [anasse.hanafi94@mail.com](mailto:anasse.hanafi94@mail.com)

---

## 1. INTRODUCTION

This study focuses on the contribution of computer vision in human-machine interfaces (HMI) applied to augmented reality (AR) through the discovery and improvement of different interfaces. The aim is to create a powerful synergy between technologies and non-computerized day-to-day activities by blurring the barrier between them [1]. There has been rapid innovation in interfaces recently due to the boom of mobile, tablets and laptops. Hence, the availability of different mechanisms for recognition capabilities and enhanced user experience through various mobile applications. This includes computer vision, speech, automatic generation of texts [2] realistic images, and video.

In this paper, we study the use of computer vision as an innovative example of AR applications. Our goal is to facilitate the learning of higher studies students by using innovative educational mobile applications. The proposed framework consists primarily in detecting objects used during the realization of a chemistry experiment, through a transformer-based model called detection transformer (DETR); then as a second step, to predict the name of the experiment using a deep learning technique, more precisely, predicting through a multi-class classification model.

## 2. RELATED WORK

Object detection is a very active area of research that seeks to classify and locate regions/areas of an image or video stream. This field is at the crossroads of two others: image classification and object localization. In this section, we shall have an overview of the different models of neural networks used for detecting objects in images.

### 2.1. Neural networks

Artificial neural networks are highly connected networks of elementary processors operating in parallel. Each elementary processor calculates a unique output based on the information it receives. Any hierarchical structure of networks is a network [3]. This definition explains that each neuron is a basic function. A neuron can, for example, receive a number, multiply it by two, and then send it to the following neurons. Neural networks are often organized in several layers, as in Figure 1. This makes it possible to better organize the path of information and facilitates the learning of the model.

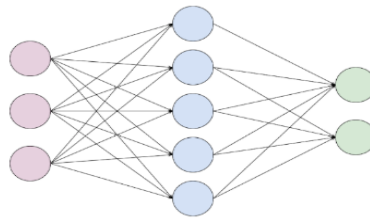


Figure 1. Architecture of a three-layer neural network

#### 2.1.1. Convolutional neural network

The convolutional neural network (CNN) is comparable to a standard neural network. It is also composed of neurons with different simple operations [4]. However, CNNs assume that the incoming data is in the form of a matrix. This type of data appears in several areas: audio wavelength [5], natural language processing [6], or more commonly, images [7]. Assuming a matrix shape for the data, allows CNN to create a layer that transforms it from a sliding window. These convolutional layers make it possible to compact the information of a region into a single piece of data.

#### 2.1.2. Region-based convolutional neural network

Region-based convolutional neural network (R-CNN) is an object detection architecture [8]. The R-CNN starts by extracting interesting regions from the image; then it uses these regions as input data for a CNN. This separation into regions makes it possible to detect several objects of several different classes in the same image. In the R-CNN presented in Figure 2, the regions are extracted due to a selective search proposed by [9]. This uses the structure of the image and several partitioning techniques to recover all possible regions of interest. This solution proposed by [10] has improved the accuracy of detection models.

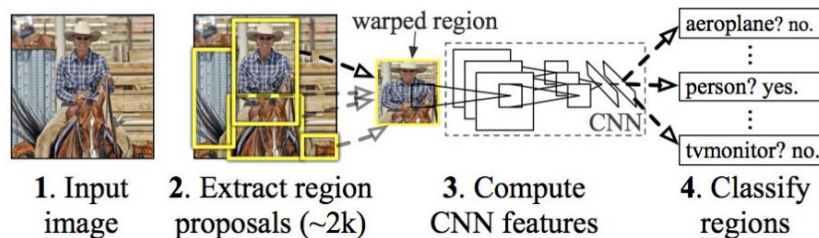


Figure 2. Regions with CNN features

#### 2.1.3. Faster R-CNN

The Faster R-CNN is an improvement of the R-CNN [11]. The architecture uses the same feature maps resulting from the convolution layers to generate the regions of interest and then classify them. The region proposal network uses sliding windows of different sizes and ratios to analyze the feature map in depth. These changes significantly improve the accuracy and speed of the architecture compared to R-CNN.

#### 2.1.4. Mask R-CNN

Mask R-CNN is the next enhancement after Faster R-CNN. The R-CNN Mask changes the output of the final model. Indeed, the Faster R-CNN architecture allows to locate distinct objects with a bounding box. The Mask R-CNN architecture makes it possible to segment each instance of an object with a semantic mask. This improvement therefore allows the model to perform instance segmentation [12].

As it can be seen in Figure 3, the Mask R-CNN architecture is similar to Faster R-CNN but adds layers in parallel to the classification. This approach is different from other instance segmentation architectures that classified the results against the generated mask [13]. This parallel approach allows each type of object to predict its mask without being in competition with the other classes of the model.

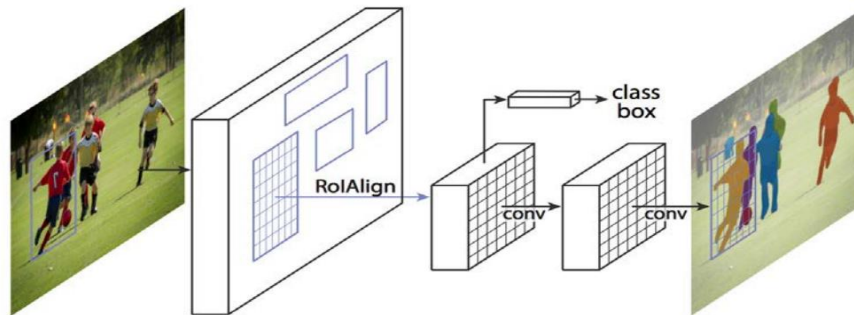


Figure 3. The Mask R-CNN framework for instance segmentation

#### 2.1.5. ResNet

Residual networks (ResNet) are neural networks that implement residual learning [14]. This innovation facilitates the learning of deep neural layers. Residual learning stems from the observation that the deep layers of a neural network had difficulty converging. Residual learning allows information to bypass certain layers and thus reduces the problem of degradation.

### 2.2. Detection transformer

As seen in the previous sections, current deep learning algorithms perform multi-step object detection. The problem of near duplicates is not avoided and impacts the accuracy of the detection [15]. It is for this reason we have chosen to use the new DETR model in our contribution. An innovative and effective approach with a higher accuracy detection. This new model designed by Meta AI researchers processes the object detection task as a direct-set prediction using a transformer-based encoder-decoder architecture. Note that transformers are the new generation of deep learning models that have achieved outstanding performance in the field of natural language processing (NLP).

#### 2.2.1. Architecture

The overall architecture of DETR models is simple; it is composed of three main components: i) CNN, ii) encoder-decoder transformer, and iii) feed-forward network (FNN). Here, the CNN backbone generates a feature map from the input image. Then the output of the CNN backbone is converted into a one-dimensional feature map which is passed to the transformer encoder as input. The output of this encoder is a number  $N$  of fixed-length integrations (vector), where  $N$  is the number of objects in the image, predicted by the model.

An encoder-decoder type model [16] has a particular architecture. It is made up of two parts, an encoder part and a decoder part. The encoder part takes the input from the model and transforms it into vectors representing information about it. In the case of an image, this part has the effect of reducing its size. The second part, the decoder, takes these vectors as input to transform them into the desired output. In the case of an image, this will enlarge the size of the vectors to transform them back into an image of a suitable size. To achieve this enlargement, information from the encoder, when the image was larger, is reused in the decoder. This is possible due to a process called skip-connections [17] which creates a link between two layers of the network. In Figure 4, we observe that the size of the initial image is reduced, due to pooling operations [18]. Conversely, in the decoder, enlargement operations increase the size of the image. Due to the arrows in Figure 4, we can clearly see that the enlargement operations use the information from the encoder. Theoretically, each of these parts can operate independently.

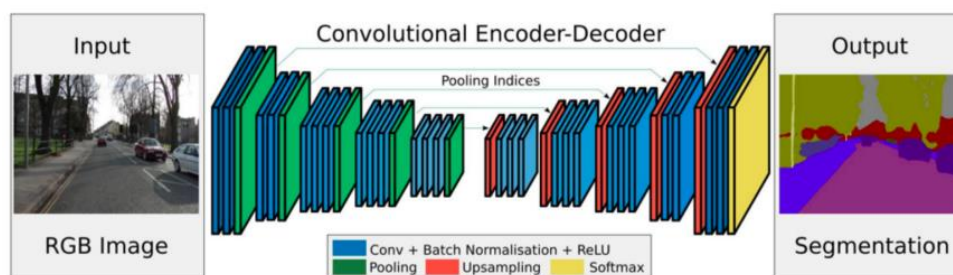


Figure 4. Segmentation of a road scene image using convolutional encoder-decoder architecture

### 3. METHOD

In a world where new technologies are increasingly present in our daily lives, AR is continuously gaining ground as a rich and versatile interface. Our contribution in this paper, seeks to integrate the use of machine learning algorithms (MLA), in an educational mobile application. Educational experiments in chemistry are a good example of the use of augmented reality. It is possible to enrich the scene of a chemistry laboratory with virtual objects such as text, images, audio, and animation. Our goal is to develop a mobile application capable of proposing the name of a chemistry experiment through two steps, as a first step, an input image of a chemistry experiment is provided by the camera of the mobile, then, by using DETR model, the application detects the tools used during the realization of the experiment. As a second step, through a multi-class classification, we predict the name of the experience based on detected objects.

#### 3.1. Proposed architecture

Figure 5 shows the overall architecture of the proposed system. The latter can be divided into two parts. The first part based on the DETR model allows detection of objects from the input image. The convolutional neural network works as the second part of the system, taking detected objects as an input from the previous part to predict the name of a chemical experiment by performing a multi-class classification.

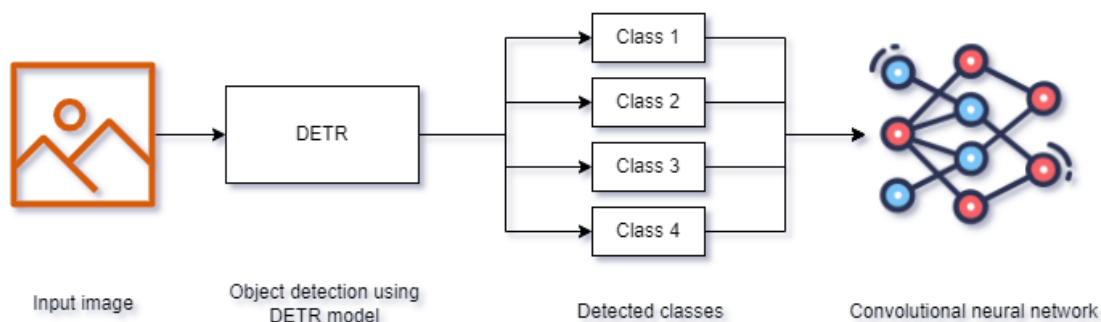


Figure 5. Overall architecture of the proposed system

#### 3.2. Object detection

The baseline DETR model developed by Meta was trained on the common object in context (COCO) dataset [19]. The latter was created by Microsoft with the aim of advancing the state of the art in object recognition. It consists of images of complex daily scenes containing common objects in their context. Regarding our contribution, we are interested in laboratory equipment such as: round bottom flask, thermometer, eye dropper, separatory funnel, Bunsen burner, test tube, beaker, watch glass, support, and Erlenmeyer flask. The COCO dataset does not contain such objects.

##### 3.2.1. Dataset

Collecting a dataset might seem like an easy task that can be done in the background while spending most of the time and resources into building the machine learning model. However, as practice shows [20], dealing with data might take most of the time due to the sheer scale that this task might grow to. It is important to understand how a dataset is composed, how it was annotated, and what features it has. Table 1 summarizes the content of our collected dataset.

Table 1. Predictable class provided by the new dataset

Objects	Number of images	Size in disk (Mb)
Round bottom flask	460	61.1
Thermometer	121	11
Eye dropper	308	30.3
Separatory funnel	159	17
Bunsen burner	172	18
Test tube	195	22
Beaker	170	18.6
Watch glass	163	16.7
Support	132	12.4
Erlenmeyer flask	995	114
Thin layer of aluminum	223	25.4

The annotation file is formatted in the JavaScript object notation (JSON) and is a collection of: info, licenses, images, annotations, and categories as shown in Figure 6. The info section contains high level information about the dataset. Licenses section contains a list of image licenses that apply to images in the dataset. Images section contains the complete list of images in the dataset; there are no labels, bounding boxes, or segmentations specified in this section; it is simply a list of images and information about each one. Categories section contains a list of categories (e.g., cat, bicycle) and each of those belongs to a super category (e.g., animal, vehicle); the original COCO dataset contains 90 categories. Annotations section contains a list of every individual object annotation from every image in the dataset. For example, if there are 70 round bottom flasks spread out across 100 images, there will be 70 round bottom flasks annotations (along with a ton of annotations for other object categories). Often there will be multiple instances of an object in an image. Usually this results in a new annotation item for each one. Figure 7 shows an extract of different images from the dataset we have collected. 80% of images are used as a training dataset; 15% are used as a validation dataset and 5% are used as a testing dataset.

```
{
  "info":{
    "description":"Chemistry tools dataset", "url":"http://fati.ac.ma/datasets/chemistry-tools","version":"1.0", "year":2020,
    "contributor":"Abdelmalek Essaadi University", "date_created":"2020/09/15"
  },
  "licenses":{
    {
      "id":1,
      "url":"http://creativecommons.org/licenses/by-nc-sa/2.0/",
      "name":"Attribution-NonCommercial-ShareAlike License"
    }
  },
  "categories":{
    {"supercategory":"Chemistry utensils", "name":"Erlenmeyer flask", "id":1},
    {"supercategory":"Chemistry utensils","name":"Round bottom flask","id":2},
    {"supercategory":"Chemistry utensils","name":"Beaker","id":3},
    {"supercategory":"Chemistry utensils","name":"Test tube","id":4}
  },
  "images": [
    {"file_name": "00--Erlenmeyer_flask/Erlenmeyer_flask_1.jpg", "height": 433, "width": 650, "id": 1},
    {"file_name": "00--Erlenmeyer_flask/Erlenmeyer_flask_2.jpg", "height": 857, "width": 937, "id": 2},
    {"file_name": "00--Erlenmeyer_flask/Erlenmeyer_flask_3.jpg", "height": 742, "width": 1024, "id": 3},
    {"file_name": "00--Erlenmeyer_flask/Erlenmeyer_flask_4.jpg", "height": 683, "width": 1023, "id": 4},
    {"file_name": "00--Erlenmeyer_flask/Erlenmeyer_flask_5.jpg", "height": 683, "width": 1023, "id": 5}
  ],
  "annotations": [
    {"id": 1, "bbox": [131.0, 212.0, 106.0, 179.0], "segmentation": [[[131, 334], [237, 334], [237, 391], [131, 391]]], "image_id": 1,
    "ignore": 0, "category_id": 1, "iscrowd": 0, "area": 18974.0},
    {"id": 2, "bbox": [334.0, 214.0, 363.0, 632.0], "segmentation": [[[334, 214], [697, 214], [697, 846], [334, 846]]], "image_id": 2,
    "ignore": 0, "category_id": 1, "iscrowd": 0, "area": 229416.0},
    {"id": 3, "bbox": [0.0, 336.0, 195.0, 323.0], "segmentation": [[[0, 336], [195, 336], [195, 659], [0, 659]]], "image_id": 5,
    "ignore": 0, "category_id": 1, "iscrowd": 0, "area": 62905.0},
    {"id": 8, "bbox": [120.0, 127.0, 84.0, 148.0], "segmentation": [[[120, 127], [204, 127], [204, 275], [120, 275]]], "image_id": 8,
    "ignore": 0, "category_id": 1, "iscrowd": 0, "area": 12432.0},
    {"id": 16, "bbox": [36.0, 41, 198, 375], "segmentation": [[[36, 41], [234, 41], [234, 416], [36, 416]]], "image_id": 15, "ignore": 0,
    "category_id": 1, "iscrowd": 0, "area": 74250.0}
  ]
}
```

Figure 6. An extract of the dataset annotation file



Figure 7. An extract of images from the collected dataset



### 3.3. Multi-class classification

The second part of the system consists in building a neural network to predict the name of the chemistry experiment shown in the input image based on detected classes by the first part of the system. We created our dataset based on the experiments shown in Table 2. For each experiment, if an object “or class” from Table 1 is used during the realization of the experiment, then the value 1 is assigned, otherwise 0. The results were stored in a comma-separated values (CSV) file and are presented in Figure 8.

Table 2. List of chemistry experiments by description

Name	Description
Water recognition test	Determines whether the substance used contains water
Starch recognition test	Determines whether the substance used contains starch
Glucose recognition test	Determines whether the substance used contains glucose
Highlighting of acidity	Determines the acidity of a chemical substance
Chemical species separation	Allows the separation of substances that constitute an element
Chromatographic analysis	Chromatography is a laboratory technique for the separation of a mixture into its component

```

1,0,0,0,0,0,0,0,1,0,0,0,water-recognition-test
0,0,0,1,0,0,0,1,0,0,0,0,Starch-recognition-test
0,0,0,1,0,1,1,1,0,0,0,0,glucose-recognition-test
0,0,0,1,0,1,0,1,0,0,1,0,glucose-recognition-test
0,0,0,1,0,0,1,0,0,0,0,0,highlighting-of-acidity
0,0,0,1,0,0,0,1,0,0,0,0,highlighting-of-acidity
0,0,0,1,0,0,0,0,1,0,0,0,highlighting-of-acidity
0,0,0,0,1,0,0,0,0,1,1,0,chemical-species-separation
0,1,1,0,0,1,1,0,0,1,0,0,chemical-species-separation
0,0,0,0,0,0,0,0,1,0,0,0,1,chromatographic-analysis

```

Figure 8. The first ten lines of the dataset used for the neural network

The first 12 columns of the CSV file represent an object that can be detected by the first part of the system. The last column contains the name of a chemical experiment. The format of the CSV file is as: spatula, round-bottom-flask, thermometer, eye-dropper, separatory-funnel, Bunsen-burner, test-tube, beaker, watch-glass, support, Erlenmeyer-flask, thin-layer-of-aluminum, and experience-name. We duplicated and shuffled collected data in order to obtain a large number of records (about 500 lines) in the CSV file.

The topology as shown in Figure 9 of the neural network is a fully connected network with one hidden layer that contains 24 neurons. The input layer contains 12 neurons, while the output layer creates 6 output values, one for each predictable experience. The output value with the largest value will be taken as the class predicted by the model. Note that we use a SoftMax activation function in the output layer, this is to ensure the output values are in the range of 0 and 1 and may be used as predicted probabilities. Finally, the network uses the efficient AdamW gradient descent optimization algorithm with a logarithmic loss function called “categorical\_crossentropy”. We have trained the model on 200 epochs and set the batch size to 10.

```

model = Sequential()
model.add(Dense(24, input_dim=12, activation='relu'))
model.add(Dense(6, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

```

Figure 9. Topology of the proposed neural network

## 4. RESULTS AND DISCUSSION

### 4.1. Detection transformer

DETR usually requires an intensive training schedule. We trained our model on a single virtual machine with 8 GPU for 300 epochs. The average time to train the model for one epoch is 30 minutes. The total training time took about 6.25 days. We train the DETR model with AdamW optimizer setting the learning rate in the transformer to 1e-4 and 1e-5 in the backbone. Horizontal flips, scales and crops are used for augmentation. Images are rescaled to have min size 800 and max size 1,333. The transformer is trained with a dropout of 0.1, and the whole model is trained with a grad clip of 0.1.

To assess the performance of our baseline model, we have calculated different metrics. Figure 10 shows the obtained results. The average precision (AP) is used to determine the accuracy of a set of object detections from the model when compared to ground-truth object annotations from the dataset. The intersection over union (IoU) [21] is used when calculating AP. It is a number from 0 to 1 that specifies the amount of overlap between the predicted and ground-truth bounding box. We provide two resulting images as shown in Figure 10 of chemistry laboratories showing the objects detected by the model.

```
IoU metric: bbox
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.395
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.603
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.414
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.175
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.430
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.591
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.323
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.516
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.559
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.284
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.615
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.807
```

Figure 10. Precision and recall metrics calculated during epoch number 150

The DETR paper [22] notes four characteristics to compare to Faster R-CNNs: i) architecture simplicity, ii) improved performance for detection of larger objects, iii) improved precision, and iv) predicting objects in parallel. One of the major advantages of DETR is the elimination of specific designed components [22] such as anchor generation and non-maximum suppression (removing duplicate regions) and that is due to the combination of convolutional network backbones and encoder-decoder transformers. Also, transformers have based its main idea on a self-attention mechanism that has reduced the number of computations relating to input/output symbols to only  $O(1)$  [23], allowing to model dependencies regardless of their position in sequence.

According to calculated metrics as shown in Figure 10 and obtained detection results as shown in Figure 11, our model achieves high detection performance; but suffers from slow convergence compared to previous detectors [24]. In fact, our model requires 300 epochs, while the Faster R-CNN [11] needs only 37 epochs for training. To ease this issue, [25] proposes a deformable detection transformer, which introduces deformable attention to mitigate the slow training process. It should be noted that the Faster R-CNN has been equipped with multiple design improvements since the baseline publication. DETR is a relatively new model and there is still space for potential improvements.



Figure 11. Detected chemistry tools shown by the green bounding box

**4.2. Multi-class classification**

Classification problems having multiple classes with imbalanced datasets present a different challenge than a binary classification problem [26]. The skewed distribution makes many conventional machine learning algorithms less effective, especially in predicting minority class examples. Imbalanced datasets refer to a classification problem where classes are not represented equally. In order to mitigate this problem, we redefined our dataset to represent all classes equally; we removed samples from over-represented classes and added more samples from under-represented classes. In order to evaluate the performance of the proposed neural network, we have logged the values of loss and accuracy functions. Figure 12 represents the evolution of those two functions over 200 epochs.

The loss function allows us to know how the model behaves on test and validation datasets. Technically, it is the sum of the errors identified in that dataset. The accuracy metric allows us to assess the performance of the model. Technically, it is the number of correct predictions divided by the total number of predictions. The results are summarized as both the mean and standard deviation of the model accuracy on the dataset. This is a reasonable estimation of the performance of the model on unseen data. It is also within the realm of known top results for this problem. As shown in Figure 12, our model achieves approximately 91% efficiency while the error estimate decreases to 12%.

We present the below results obtained as shown in Figure 13 when using the model on our prepared dataset. Assuming that the first part of the system detected the following objects: round bottom flask, thermometer, separatory funnel, Bunsen burner, and watch glass. The second part of the system gets those objects as an input. While running the model, we print for each predictable experience its corresponding vector. The resulting vector of the model will represent the prediction for each experience. The output with the largest value will be taken as the class predicted by the model. The example in Figure 13, shows that the second value of the resulting vector is the large one and corresponds to chemical species separation experience. This prediction is correct; the objects detected by the first part of the system have been used to separate the substances that constitute a chemical solution.

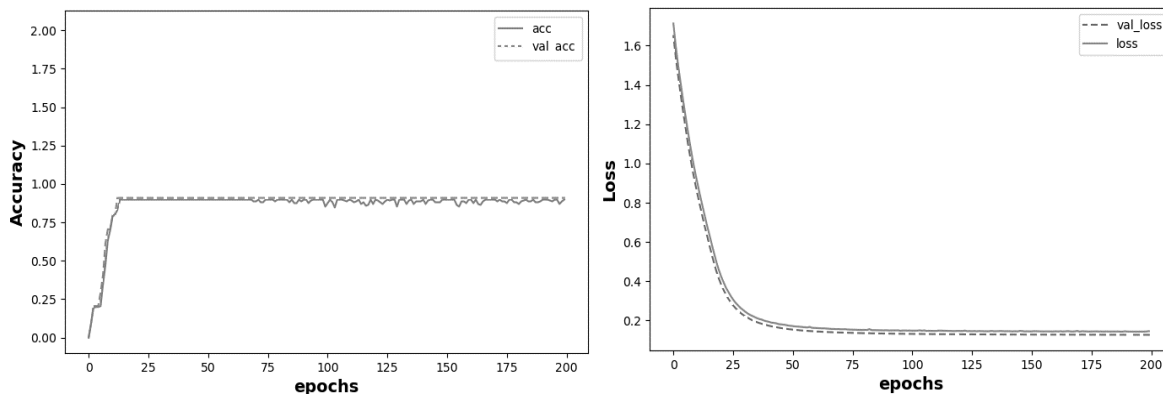


Figure 12. Accuracy and loss metrics calculated on training and validation dataset

```

Detected objects :
Round bottom flask; Thermometer; Separatory funnel; Bunsen burner; Watch glass

Corresponding input vector :
[0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0]

Corresponding vectors for each predictable experience:

water-recognition-test      =====> [0. 0. 0. 0. 0. 1]
Starch-recognition-test     =====> [1. 0. 0. 0. 0. 0]
glycose-recognition-test    =====> [0. 0. 0. 1. 0. 0]
highlighting-of-acidity     =====> [0. 0. 0. 0. 1. 0]
chemical-species-separation =====> [0. 1. 0. 0. 0. 0]
chromatographic-analysis    =====> [0. 0. 1. 0. 0. 0]

Resulting vector :

[[1.2165614e-05 9.9102950e-01 7.9927107e-05 8.3226012e-03 3.2622780e-04
 2.2960413e-04]]
    
```

Figure 13. Output result obtained when running the deep learning model






## 5. CONCLUSION

The adoption of computer vision techniques in both the learning and teaching process is opening new perspectives for the use of augmented reality and deep learning. Computer aided learning (AR and virtual reality training) helps students to understand complex concepts, thus improving their efficiency. For many computer-vision tasks, neural networks, more specifically deep convolutional neural networks have become the reference technique. DETR is an exciting phase forward in the world of object detection. It features an important reduction in priors and a simple, easy-to-configure network architecture. It outperforms Faster R-CNN in most tasks without much particular extra work; however, it is still slower than comparable single-stage object detectors. Its simple structure makes it easy to recreate, experiment with, and finetune from the strong baseline provided by the researchers. Furthermore, transformers have multiple implementations, especially in computer vision, but there is still space to propose more generic designs. In this paper, the obtained results of the proposed system have proved that combining multiple deep learning and computer vision techniques is possible and promising.





## REFERENCES

- [1] R. Silva, J. C. Oliveira, and G. A. Giraldo, "Introduction to augmented reality," *National Laboratory for Scientific Computation*, vol. 11, no. 1–11, 2003, doi: 10.1007/978-1-4842-7462-0\_2.
- [2] A. Hanafi, M. Bouhorma, and L. Elaachak, "Machine learning-based augmented reality for improved text generation through recurrent neural networks," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 2, pp. 518–530, 2022.
- [3] C. Touzet, "Les réseaux de neurones artificiels, introduction au connexionnisme," *Collection de l'EERIE*, 2016.
- [4] "CS231n convolutional neural networks for visual recognition," GitHub, 2022. <https://cs231n.github.io/convolutional-networks/> (accessed May 24, 2022).
- [5] J. Lee, T. Kim, J. Park, and J. Nam, "Raw waveform-based audio classification using sample-level CNN architectures," *arXiv:1712.00866*, Dec. 2017.
- [6] M. M. Lopez and J. Kalita, "Deep learning applied to NLP," *arXiv:1703.03091*, Mar. 2017.
- [7] I. Goodfellow and Y. Bengio and A. Courville "Deep learning," MIT Press. <https://www.deeplearningbook.org/> (accessed May 25, 2022).
- [8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.
- [9] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.
- [11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv:1703.06870*, Mar. 2017.
- [13] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 3150–3158, doi: 10.1109/CVPR.2016.343.
- [14] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *ECCV*, 2020. Accessed: May 27, 2022. [Online] Available: <https://paperswithcode.com/paper/end-to-end-object-detection-with-transformers>
- [16] J. C. Ye and W. K. Sung, "Understanding geometry of encoder-decoder CNNs," *arXiv:1901.07647*, Jan. 2019.
- [17] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: on the transferability of adversarial examples generated with ResNets," *arXiv:2002.05990*, Feb. 2020.
- [18] C.-Y. Lee, P. Gallagher, and Z. Tu, "Generalizing pooling functions in CNNs: mixed, gated, and tree," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 863–875, Apr. 2018, doi: 10.1109/TPAMI.2017.2703082.
- [19] T.-Y. Lin *et al.*, "Microsoft COCO: common objects in context," May 2014, *arXiv:1405.0312*.
- [20] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-AI integration perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, Apr. 2021, doi: 10.1109/TKDE.2019.2946162.
- [21] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 658–666, doi: 10.1109/CVPR.2019.00075.
- [22] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12346, Springer International Publishing, 2020, pp. 213–229.
- [23] M. Chromiak, "Exploring recent advancements of transformer based architectures in computer vision," in *Selected Topics in Applied Computer Science*, 2021, pp. 59–75.
- [24] Z. Sun, S. Cao, Y. Yang, and K. Kitani, "Rethinking transformer-based set prediction for object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 3591–3600, doi: 10.1109/ICCV48922.2021.00359.
- [25] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: deformable transformers for end-to-end object detection," *arXiv:2010.04159*, 2020.
- [26] M. Joo Er, R. Venkatesan, and N. Wang, "An online universal classifier for binary, multi-class and multi-label classification," in *IEEE Int. Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2016, pp. 3701–3706, doi: 10.1109/SMC.2016.7844809.





**BIOGRAPHIES OF AUTHORS**

**Anasse Hanafi**     received his engineering degree in computer science on 2017 from the Faculty of Sciences and Technologies, University Abdelmalek Essaâdi, Tangier, Morocco. He is currently a Ph.D. candidate, his research interest's augmented reality (AR). He has published several research articles in international conferences of computer science. He can be contacted by email: [anasse.hanafi94@gmail.com](mailto:anasse.hanafi94@gmail.com).



**Lotfi Elaachak**     is a professor (Assistant) at Abdelmalek Essaâdi University since 2018, He has published several research articles in international conferences of computer science. His research interests include decision trees, genetic algorithm, machine learning, neural networks, artificial intelligence, genetic programming and many more. He can be contacted by email: [lotfi1002@gmail.com](mailto:lotfi1002@gmail.com).



**Mohammed Bouhorma**     is an experienced academic who has more than 25 years of teaching and tutoring experience in the areas of Information Security, Security Protocols, AI, Big Data and Digital Forensics at Abdelmalek Essaadi University. Bouhorma received his M.S. and Ph.D. degrees in Electronic and Telecommunications from INPT in France, He has held a Visiting Professor position at many Universities (France, Spain, Egypt and Saudi Arabia). His research interests include, cyber-security, IoT, big data analytics, AI, smart cities technology and serious games. He is an editorial board member for over a dozen of international journal and has published more than 100 research papers in journals and conferences. He can be contacted by email: [bouhorma@gmail.com](mailto:bouhorma@gmail.com).