Video classification of Odors with Convolutional Neural Networks

José Maria Lopes Rafael Ledesma Frazão
jose.l.frazao@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

Abstract

Trying to mimic natural olfactory systems, by achieving the perfect combination between chemical sensing and artificial intelligence, has been an important task among the science community. Over the last decade, e-nose technologies have undergone important developments on their way to be implemented in real life situations such as food monitoring, air quality monitoring, etc. In order for that to be possible, e-noses need to possess a very efficient and highly accurate pattern recognition component. In this work we will investigate how CNN based algorithms can be implemented in these devices in order to help them achieve good results. With deep learning being a hot topic nowadays, and with the advance in computational hardware, plenty of very good architectures have emerged that can achieve outstanding results in pattern recognition tasks. Different architectures are approached, from simple 3 dimension CNNs to a object detection algorithm (YOLO) and recurrent algorithms capable of handling sequencial data (LSTM). Since this work focuses solely on the pattern recognition side of e-noses, it will work on top of an already produced dataset. At first, an hybrid gel was exposed to 11 different volatile organic compounds and its reaction to them was recorded in the format of image sequences. The objective is to find the algorithm capable of classifying the image sequences corresponding to the the 11 VOC classes with the highest possible score. On the second task, this gel was exposed to the same compound but with different concentrations in the exposed gas. Thus, the second goal is to find the model best capable of predicting the actual exposed concentration, also based on the recorded image sequences.
Keywords: E-nose, CNN, LSTM, YOLO, VOC, Object Detection, Concentrations.

I. Introduction

A. Motivation

The olfactory sense is considered to be the oldest out of all senses. This is a sense that even bacteria have in order to react and adapt to the chemicals around it. Some of these bacteria were already around ages before creatures with the ability of hearing, touching or seeing started to exist [1]. On a more actual context, lots of different animals, not so much humans, still put too much trust on their smelling sense since it is used to find food, avoid danger, track their mates and their preys, etc.. This sense plays a vital role in their existence and capacity to understand the surrounding environment.

This unique capacity of detecting different smells/odors is continuously being a subject of study and plenty of advances have been made in this field. One of the ideas that came out from it was the possible development of some kind of artificial olfactory system able to identify substances in the air. Having the possibility of mimicking this highly complex and powerful system in various situations can be game changing. Electronic based devices are being developed to serve this exact purpose, which are called E-noses. These devices contain an array of sensors that, in contact with VOCs, generate signals that can later be analyzed in order to characterize and identify these same compounds. The odors present in the air are composed by mixtures of these VOCs. E-noses aim to be the perfect combination between chemical sensing, signal processing and pattern recognition inside the brain, in order to achieve high performances in odor recognition tasks.

Several of these devices are already being successfully put into practice. E-noses are already being implemented in several areas such as disease detection [2, 3, 4], quality control in laboratories [5], food quality control [6], air quality monitoring [7], security systems [8] and many more.

B. Objectives

Over the past few decades, the development in the field of AI has been astronomical. All the hype around it and perception of its immense potential led loads of people to spend their time and money towards investigation in this field. This, together with the advance in computing hardware, such as GPU, enabled fields like Machine Learning (ML) to rise and offer a whole new set of solutions to problems that once seemed unsolvable. The whole idea behind ML is to teach a computer how to perform a certain task without having the need to provide exact instructions every step of the way. Basically ML algorithms learn the underlying relations among data to then make decisions without being programmed to. One of the ways to do that is through the use of ANNs which are based on the biological neural networks that compose animal brains.

These networks later evolved to new sub-fields of ML like Convolutional Neural Networks (CNN) that managed to tackle a new set of problems with great performances. CNNs have been proven to be a very successful neural network to process image data. They have shown state-of-the-art results in many image related tasks such as recognition, detection and segmentation. CNNs are a hot topic inside the ML community that is using it for every imaginable task possible. CNNs have shown very good results in problems like Facial Recognition [9], Action Recognition [10], Cancer Detection [11] and many more.

The pattern recognition in a real olfactory system is performed by the brain that takes as inputs the electrical

signals generated when VOCs bind to the chemoreceptors in the nose. The brain has a huge pattern recognition capacity as it can easily match stimulus produced by the outside world with data stored in memory. This work focuses on the pattern recognition section of the olfactory system, not really targeting the stimulus acquisition and transducing parts. In an artificial olfactory system, many different methods can work as this pattern recognition element but in this work, CNN based systems will serve this purpose. So, the main objective of this work is to investigate how can these systems be implemented in an E-nose system by making use of their pattern recognition capabilities. CNNs have already been used in E-nose systems to predict odor pleasantness [12] or to identify a specific type of tea [13] for example. However, these works do not use CNN as feature extractors in image sequences, but in other types of signals. In both these works, the reactions to the odors were captured by Metal-Oxide and/or Quartz Microbalance sensors that record data in only one dimension. Two dimensional CNNs were still used but, in order for that to be possible, several sensor responses were put together into 2D matrices or a response was multiplied by its transpose to again obtain data in two dimensions.

The Biomolecular Engineering Group, UCIBIO, REQUIMTE, FCT/UNL developed sensors (gas-sensitive gel) that generate a distinct optical response according to the VOC they are being exposed to. This optical response can be filmed, and through its pattern variations, in theory, it is possible to identify the exposed VOC (fingerprint). In order to do that, CNN based systems will be applied to these recorded responses (image sequences).

This work can be divided into 2 different parts. The first one consists in trying to predict which VOC the gas-sensitive gel is being exposed to. On the second one, the goal is to predict the concentration of a specific VOC (Acetone) to which the gel is being exposed. For both these experiences, the results will be achieved always resorting to CNN based systems that take as input the recorded reactions of the gel (more specifically, the reaction of the droplets present in this gel). Another goal, besides trying to find the models that work better on solving these tasks, is to check whether the size of these droplets has influence on the results.

II. Theory

A. Convolutional Neural Network (CNN)

A CNN is a special type of Neural Network usually applied to computer vision tasks. CNN based algorithms have been showing great performances in several image related tasks, such as image classification, segmentation or detection, where they are able to produce state-of-the-art results.

Usually, a CNN is formed by 3 main types of layers. These are the Convolutional Layers, the Pooling Layers and the Fully Connected Layers. A typical CNN is normally composed by alternating convolutional and pooling layers, finishing with at least a fully connected one (Figure 1).

A convolutional layer is usually composed by several kernels/filters, each one useful for different analysis of nearby pixels. The fact that the set of weights in a kernel is shared

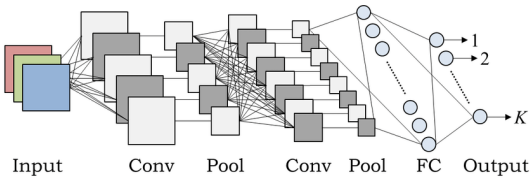

Fig. 1: Typical CNN architecture (taken from [14])

for the whole image is a huge advantage compared to a simple MLP since the amount of parameters to train decreases drastically. Besides that, if a certain feature changes its location in the image, it can still be detected given this all round spatial property. The convolution is an operation where the kernel slides horizontally and vertically over the entire image. In each position, the sum of every multiplication of a kernel element by its corresponding element in the image is stored in an output image/feature map. The dimensions of this layer's output image will depend on the original image size, on the kernel size, the stride applied on the convolution, padding, etc.. As soon as the convolutions of every kernel are done, a feature map is generated for each one of them.

Normally, every convolutional layer goes hand in hand with an activation function. Every produced value by the convolutional operation is still subject to a non linear function known as activation function. Without signal activation, the output signal of this layer would merely be a linear function. In this scenario, the neural network would become a simple linear regression model that would not be able to handle complex data and would not perform well enough most of the times. Knowing this, adding non-linearities to the system makes it more robust and capable of tackling more difficult problems. The most popular activation functions nowadays are the Sigmoid, Tanh and ReLu. The last one has been showing better performances compared to the others and it is the most used nowadays [15].

The second main type of layers used in CNNs are the Subsampling/Pooling Layers. The main reason for using these types of layers is to reduce the system dimensionality, which consequently reduces the amount of parameters to train in the process. To do that, it combines the values of a certain region in the feature map into a single value. This operation sums up the information in this region, outputting one single value, according to the used function. This is also done using the sliding window technique. By decreasing the dimensionality of the network, the amount of parameters to train decreases which can help avoid overfitting.

Another also very commonly used layer in CNNs is the Fully Connected Layer. These layers usually appear at the end of the CNN. They work as a traditional MLP where every neuron in the previous layer is connected to every neuron in this layer. Their goal is to take the feature maps outputted by the previous layers, flattened, and predict the final result. In a classification task, the output layer usually has the same amount of neurons as the number of classes. To assign to each class its own probability, this last layer normally takes the softmax function as its activation function. In a regression

task, where the goal is not to predict a class but a numeric score, the output layer is normally a single neuron with no activation function.

There are also regulatory layers, like Batch Normalization and Dropout layers, that help optimizing the network performance. Batch normalization layers [16] can adaptively normalize data during training, by keeping track of two additional parameters (moving average of the batch-wise mean and variance), which helps mitigating the vanishing gradient problem allowing networks to get deeper. Besides that, they also add a bit of noise to hidden layers, reducing overfitting slightly and helping in the quest for generalization. Then, Dropout layers set to zero random entries of the output feature map of the previous layer, depending on the chosen dropout rate. By setting a neuron to 0, the cost function becomes more sensitive on how neighbouring neurons change their weights during backpropagation, also helping the model to generalize.

Even knowing all this, to build a network capable of solving a certain problem with a high performance is not a straight forward task. Having these notions can help getting a starting point for a Network but the odds of it generating the best possible results at its first try are slim. In order to achieve better and better performances, one needs to get past the exhaustive and slow process of parameter tuning. The problem of this process is that sometimes to train and test one version of a network can take long periods of time and the great amount of parameters to tune does not help.

B. Recurrent Neural Network (RNN)

Unlike vanilla ANNs or CNNs, RNNs (Figure 2) are able to capture information over time. These networks can remember the past and base their current decisions on what they previously learned. They are able to do this by keeping a state containing information regarding what they have learned until a certain moment. Between different sequences this state is reset since sequences are single inputs to the network. These networks are able to capture the time dynamics of a sequence hence they can input or output sequences of data that are not independent from each other.

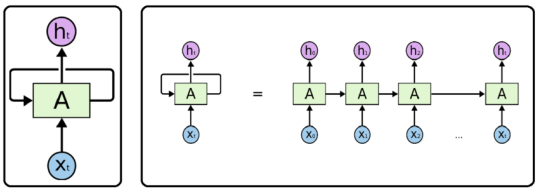

Fig. 2: Simple RNN architecture (taken from [17])

Due to the completely different architecture of RNNs compared to the previously mentioned ones, the standard method for network training can no longer be used. The recurrent loops present in these types of networks do not allow the simple backpropagation to update the weights. Backpropagation Through Time (BPTT) [18] was developed to train this type of networks but as the number of timesteps increased, the vanishing gradient problem started to emerge. A tweak to this method was made, appearing the Truncated Backpropagation Through Time (TBPTT) [19]. Although more effective in solving the vanishing gradient problem, it is not able to learn dependencies between timesteps more than a certain number of units apart. But this was not the only way found to alleviate the vanishing/exploding gradient problem. A new architecture called Long Short-Term Memory (LSTM) allowed to soften this problem and consequently to implement deeper networks capable of dealing with larger sequences.

This new model resembles a typical RNN where every hidden layer is replaced by a memory cell. Unlike a simple RNN which has a single neural network layer in its hidden block, these memory cells are composed by several of them (Gates) that control the information that flows through the sequence chain.

C. Object Detection

Object Detection is an area that falls into the Computer Vision field, which deals with the identification and localization of objects in images. This can be done either by drawing a bounding box that contains the object or by selecting every pixel in the object (segmentation).

You Only Look Once (YOLO) is one of the established object detections algorithms and has as its biggest advantage the superb speed at which it can perform this task (45 frames per second). While other object detection algorithms need to go through an image several times before being able to detect an object, YOLO [20] only has to look at it once, hence the name You Only Look Once. The YOLO developer team reframed object detection as a single regression problem capable of extracting bounding box coordinates and class probabilities straight from image pixels.

At first, an image is divided into a 7x7 grid. If the center of a certain object falls into a specific cell, it becomes responsible for detecting that same object. The problem with this approach is that any grid cell can detect only one object. This means that only a maximum of 49 objects can theoretically be detected and if a cell contains more than one object it will not be able to correctly detect all of them. These problems were solved in future versions of YOLO. Each one of the 7x7 grid cells predicts 2 bounding boxes. Every bounding box is subject to the IoU method to compared them to the ground truth boxes. Boxes with a score higher than a certain are kept and the remaining ones are rid of. The model also outputs 4 numbers that represent the location of the box relative to the bounds of its corresponding cell. To obtain the bounding box predictions and their class probabilities, YOLO uses only a single CNN.

Although the speeds obtained for the first model were incredibly fast, allowing the algorithm to be used in real time applications, the number of localization errors was significant and the network presented a relatively low recall. So to try and achieve a better performance, a new YOLO version was developed (YOLOv2) [21]. In this new architecture, batch normalization layers were added after every convolutional layer and the resolution for detection was increased. But the

main introduction was the use of Anchor Boxes which allows multi-object prediction per grid cell. The fully connected layers previously used on YOLO to predict bounding boxes were removed and replaced by anchor boxes. Anchor boxes are a set of predefined bounding boxes with certain values of height and width. They are chosen according to the typical ground truth boxes present in the dataset to use. These anchor boxes are tiled across the image, and the predicted bounding boxes are basically refinements of these original boxes.

III. Data

Since this work is focused on the pattern recognition side of an e-nose it has to work on top of an already produced dataset. The Biomolecular Engineering Group, UCIBIO, REQUIMTE, FCT/UNL developed a gas-sensitive gel which can be used for odor detection [22]. It contains droplets that, when exposed to a VOC, show a certain dyanimc pattern (Figure 3 and 4). A full exposure consists on the exposure cycle (gas gets pumped into the chamber containing the gel) and the recovery cycle (clean air removes the gas from the chamber).

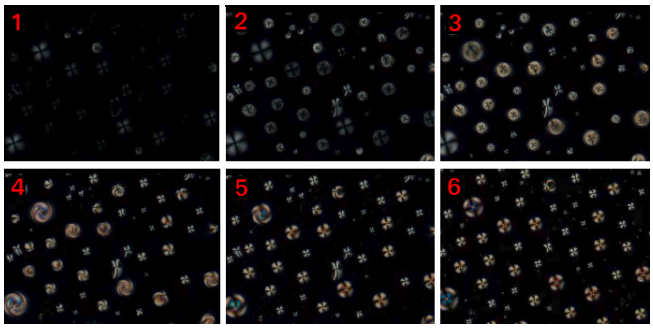

Fig. 3: Excerpt of a collected Acetone image sequence in recovery cycle

Fig. 4: Acetone droplet evolution in a full cycle

For the first task, where the goal is to identify the exposed VOC, the reactions of 11 different gels were recorded to the exposures of 11 different VOCs (Acetone, Acetonitrile, Chloroform, Dichloromethane, Diethylether, Ethanol, Ethylacetate, Heptane, Hexane, Methanol, Toluene). Each VOC was exposed to the same gel 5 times (55 total exposures).

On the second task, where the goal is to predict the concentration of the VOC exposed, 6 gels were exposed 11 times to Acetone with increasing concentrations (66 total exposures).

IV. Approach

A. General Overview

Given that the ultimate goal of this project is to develop a device capable of identifying substances/find their concentrations in the air, according to the visual reaction of small droplets over time, two main tasks have to be performed: **Detection** and **Classification/Regression**. Correctly detecting the droplets is the target for the first task and it is shared for both problems of this Thesis. The idea was to train a model (YOLO) capable of mastering droplet detection and that was able to perform this action on a very short period of time so it could eventually be implemented on a real time system. It is crucial for this model to obtain spot on detections in order to take full advantage of the models that follow. Every frame in a sequence is subject to the YOLO network and, based on a Intersection over Union method applied in every detected bounding box, the droplet sequences are built. The second task was to develop a system that, given these droplet sequences, would be able to make predictions regarding the VOCs present in the air or their concentration. Given the theme of this thesis, both these tasks were performed having CNNs at their core. For the classification/regression part several different methods were tested, such as 2 dimensional CNNs plus voting system, 3 dimensional CNNs, LSTMs with 2 dimensional CNNs as feature extractors and stacking ensembles. Figure 5 presents a scheme of the overall system architecture.

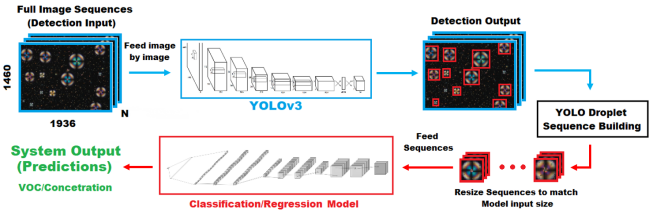

Fig. 5: Full system architecture

B. Models

Every test performed at the classification/regression stage was based on two different CNN models. The first one (Baseline Model 1) was inspired by the LeNet-5 network [23] (Fig 6). Some tweaks were applied to this model like changing all the *tanh* activation functions to *ReLU* ones, replacing average pooling layers for max pool ones, adding a Batch Normalization layer between the Convolution layers and their activation functions and adding dropout layers after these activation functions. Depending on which task this model was applied on, the last layer was either a 11 unit one paired with a softmax activation function (Classification) or a single neuron without any activation function (Regression).

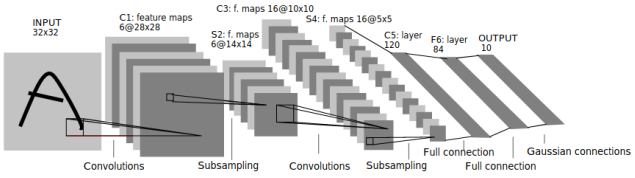

Fig. 6: LeNet-5 architecture [23]

The second CNN model (Baseline Model 2) (Figure 7), a deeper one, is composed by a total of 10 layers. The convolutional section of the model is composed by three sets of two convolutional layers followed by a max pooling one

and at the end, again, depending on what kind of problem the model is being applied on, there is a specific output layer. In this model, padding is applied in the convolutional layers in opposition to what happens with Baseline Model 1. Only one regulator layer (Dropout) was inserted in the architecture, just before the output one. For both models, no parameters were fixed since lots of variations of parameters are tested in the training stage. Also worth mentioning that both these models can be applied in a 2d or 3d setup by only changing the input size and convolutional parameters.

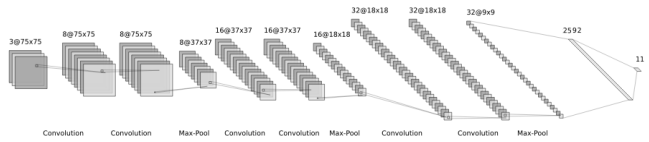

Fig. 7: CNN2D (Baseline Model 2) architecture

Given that this problem is based on time series data, at any point in the experiments, LSTMs would have to be used given their proven effectiveness on problems with these kinds of data. The type of LSTM applied was of the type Many to One since its input is a sequence of data and the output is a class/VOC or concentration, obtained through the returned values of the last cell. The amount of cells that compose the LSTM is the same as the sequence length. These are image sequences so they are not fed directly to the LSTM cells without any prior pre-processing due to the high complexity of the feature space. Instead, each image goes through a CNN2D (same for all timesteps), and the output features are then fed to the respective LSTM cells. The output returned from the last LSTM cell, that is already influenced by every timestep, is connected to the output layer. The only regularization used here is a dropout layer between the CNNs and the LSTM. The experiences that follow, besides using unidirectional LSTM, also employ BiLSTMs where the inputs are also managed from the future to the past. Instead of only one, these LSTMs have two hidden states and they use the combination of both of to produce the end result which can be beneficial given the additional data 'perspective'. Figure 8 presents a scheme on how the LSTM and CNN models work together.

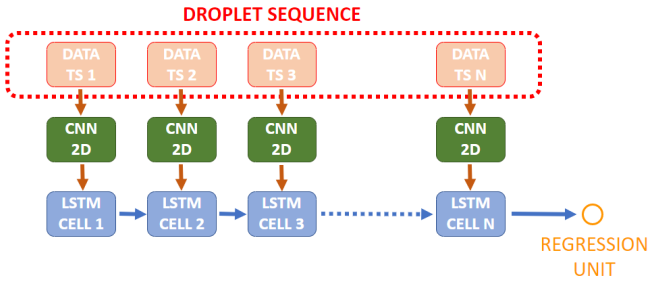

Fig. 8: LSTM + CNN2D architecture scheme

C. YOLO Droplet Sequence Building

The YOLO network produces bounding boxes around droplets on single images and not on 3D images/image sequences. In both parts of this Thesis, the goal is to produce a result based on an entire droplet sequence so a simple strategy was developed in order to build these sequences from the detections performed on all the frames. The YOLO detections on the first frame could simply extend to every other frame but this could prove to be very error prone. Instead, droplet bounding boxes are predicted for every frame. Since every droplet has a shape very similar to a circle, the bounding boxes, in theory, should all be squares in order to circumscribe them perfectly. Thus, bounding boxes with a dimension that exceeds the other by more than 20% are discarded right away. Another performed inspection is to check for overlapping bounding boxes. If two bounding boxes present an IoU score over 0.3 they are considered to be detecting the same droplet so only one can remain. The image is turned into black and white and the bounding box containing more white pixels inside it is kept since it probably contains a bigger portion of the droplet, with the other being discarded. In order to put together bounding boxes from different frames and associate them to the same droplet, it is used once again the IoU metric. Bounding boxes from the first frame are set as reference. Then, every bounding box in the remaining frames that achieves an IoU score above 0.5 with one of the reference ones is associated with it, meaning that it is very likely to be circumscribing the same droplet on a different frame. With this strategy, if a certain bounding box on the first frame does not have a very good set of coordinates, it may not be associated with other ones and a droplet sequence might be missed. However, since the first frame is the one where the droplets are still idle and in a more 'clear' form, it is in theory the frame less prone to substantial errors. After doing this association, sequences that contain very few bounding boxes, less than 25% of the sequence size, were also discarded. Finally, the mean coordinates values are computed to obtain the final ones. In order to feed these sequences to certain CNN classification/regression models, the droplets are cropped from every frame, resized to the model's input size, and put together to form a sequence. Every time a model evaluates its performance on the test set using the YOLO detections, this is the approach used to assemble the droplet sequences.

V. Experiments & Results

A. 11 VOCs Experiment

In this experiment, 11 VOCs were exposed to 11 different gels 5 times each. The 3 first exposures of every VOC were selected for the training set, the fourth exposures for the validation set and the last ones for the test set. Every exposure is kept in the form of an image sequence formed by 32 frames with 1936x1460 resolution.

1) Object Detection (YOLO): The YOLO network used to perform the droplet detection was an implementation developed by *wizyoung* made available on *GitHub* [1]. As mentioned in Section IV, the YOLO network is in charge of detecting the droplets in the gel.

[1]https://github.com/wizyoung/YOLOv3_TensorFlow

TABLE I: YOLO parameters during training

Parameters	Value
Batch Size	5
Image Resize	416x416
Keep Aspect Ratio	True
Batch Norm Decay	0.99
L2 Weight Decay	0.0005
Optimizer	Momentum
Learning Rates	0.0001, 0.00003, 0.00001
Fine Tuning	Whole Model
NMS Threshold	0.5
mAP Threshold	0.5

With the set of parameters in Table I, and after 2000 epochs of training, the model stagnated on the validation set. On this set, it was able to achieve a mAP of 0.94, a Recall around 0.95 and a Precision around 0.91. Figure 9 represents the predictions of the trained YOLO network on an image from the test set, 'unseen' by the model, on top of the ground truth bounding boxes.

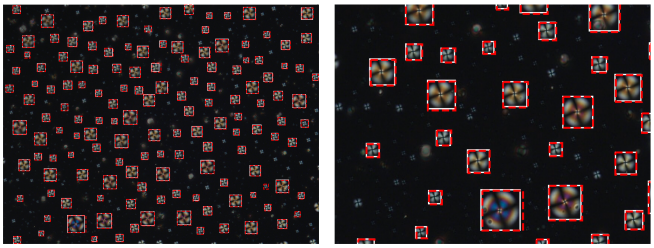

Fig. 9: Ground truth bounding boxes (White) vs YOLO detected bounding boxes (Red) - **Left**: Whole Frame **Right**: Frame Zoomed In

2) Droplet Sequence Classification: While the inputs of the YOLO network are single full frames, the models in this section will make the predictions on the VOC taking as inputs droplet sequences. Since CNN models have fixed input sizes and the droplets have a wide range of sizes, they all had to be resized to a specific dimension. At first every droplet was resized to dimensions of the biggest droplet observed in the dataset (148x148 pixels). This decision was taken since bigger droplets seem to have a more distinct dynamic pattern than smaller ones and, by doing that, they were not going to lose much resolution. On the other hand, smaller droplets would get blured. Midway through model testing, in order to speed up training, every droplet started to get resized to (75x75 pixels), since it did not seem to affect the predictions on the bigger droplets.

The models tested in order to predict the VOC associated to a droplet sequence can basically be divided in two different categories: 3 dimensional CNNs (CNN3D) and LSTMs with 2 dimensional CNNs (CNN2D) working as feature extractors. Every model was trained with the 'Adam' optimizer and using Categorical Crossentropy as the loss function.

CNN3Ds were trained for both Baseline Models, with both the original and smaller input sizes and also a One vs Rest (OvR) strategy was tested. In terms of performance on the validation set, every strategy (with exception to CNN3D BM2) was able to achieve a macro F1-score (metric selected to evaluate the models' performance) above 0.95. Even

TABLE II: CNN3D (BM1) vs CNN3D (BM1) Ensemble - F1-score on test set

	F1-Score Test (YOLO)	
VOC	**CNN3D (BM1)**	**CNN3D (BM1) Stacking**
Acetone	1	1
Acetonitrile	0.57	0.732
Chloroform	0.807	1
Dichloromethane	0.167	0.467
Diethylether	0.876	1
Ethanol	0.965	0.972
Ethylacetate	0.465	0.191
Heptane	1	1
Hexane	0.978	1
Methanol	0.951	1
Toluene	0.451	0.889
Macro	**0.748**	**0.841**

though the models worked on a level close to perfection on the validation set, when evaluated on the test set (through the YOLO generated sequences) the achieved F1-scores always took a significant dip. The best single model (CNN3D BM1) was only able to reach a macro F1-score of 0.748. Given that for each strategy many different models were trained, with different parameters, by only picking the best performing one to operate on the test set, all the remaining ones were being wasted. The best model, despite presenting the highest macro F1-score, gets outperformed by some of the remaining ones on some classes. Stacking ensemble was applied on the CNN3D trained models, as in other strategies, to try to take advantage of this situation. By training a model (meta-model) to make the final prediction based on predictions performed by many different models with different strengths, it is possible to get more accurate predictions. Different classifiers were trained (on the validation set) to step in as this meta-model. In this particular case of the 3 dimensional CNNs (BM1), the Gaussian Naive Bayes was the one achieving a better macro F1-score (0.984). The difference in performance on the test set of this ensemble compared to the performance of the single best model was significant, with an increase on the macro F1-score from 0.748 to 0.841 (Table II). The overall positive effect of the ensemble technique on the predictions was clear with every class improving its results, except for Ethylacetate that registered a significant decrease F1-score.

Regarding the effect of droplet diameters on the quality of predictions, it could clearly be observed that with an increase in size, there is also an increase in accuracy. This conclusion was taken from Figure 10 that shows a diameter based histogram, with the droplets detected on the test set, where the bin colour is associated to the accuracy in predictions of the droplets in that diameter range. While sequences of smaller droplets ($<$20µm) only achieve accuracies below 65%, with the increase in diameter, the accuracy increases with it until it reaches the 100% mark for droplets with a diameter bigger than 42µm.

After 3 dimensional CNNs, on a second phase, LSTMs paired up with 2 dimensional CNNs were tested. The main difference between these two types of networks is on how they both approach the third dimension. Again,

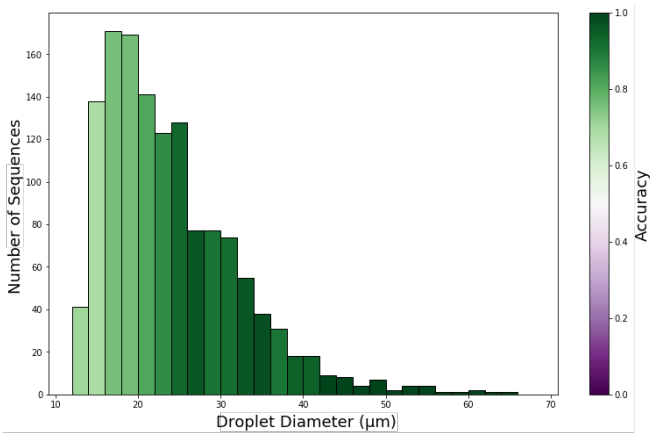

Fig. 10: Droplet Sequence Accuracy by Diameter - CNN3D (BM 1) models Stacking Ensemble - Test Set

the LSTM models were trained using both CNN models (BM1 and BM2) as feature extractors. Since the stacking ensemble worked pretty well in the first phase, it was applied again to improve the performance on single models. Every experiment using LSTMs outperformed all the ones with 3 dimensional CNNs. Even before applying any kind of stacking ensemble, single models were already achieving better results, on the test set, than the best ones obtained using CNN3D models, including the ensemble one (Table II). The best results were reached by using the 2 dimensional CNN (BM2) as feature extractor on every timestep frame with the results getting a boost when all the trained models got put together in an ensemble. The best LSTM (BM2) model got a macro F1-score of 0.908 while the ensemble was able to achieve 0.932 which is the best recorded result. Table III shows the results of this ensemble on the test set. Only the Dichloromethane and Ethylacetate classes presented an F1-score below 0.95. This happened since only 42% of the Ethylacetate sequences were correctly predicted and the remaining ones were wrongly classified as Dichloromethane. This decreases the Dichloromethane precision which is reflected in its F1-score, even though all of its sequences were correctly labeled. These mistakes in the Ethylacetate sequences happen for droplets on the lowest side of the diameter spectre (<24µm), while bigger ones are classified correctly, which helps to prove the point that smaller droplets have a lesser ability to retain the essence of a VOC.

To wrap up this 11 VOC experiment, and in order to have an overview of which strategies worked better on this task, Table IV shows the F1-scores obtained for all of them.

B. Concentrations Experiment

In this experiment, the goal is to predict the concentration of Acetone exposed to the gel. 11 different concentrations were exposed to 6 different gels. The exposures on 4 of the gels were selected for the training set, one for the validation set and other for the test set. Now, every exposure is composed by 78 frames with 1936x1460 resolution.

1) Object Detection (YOLO): Given the good results achieved on the first YOLO implementation, the same set

TABLE III: LSTM+CNN2D(BM2) Ensemble - F1-score on test set

	F1-score (YOLO)
VOC	**LSTM+CNN2D (BM2) Stacking**
Acetone	1
Acetonitrile	0.962
Chloroform	1
Dichloromethane	0.763
Diethylether	0.96
Ethanol	0.991
Ethylacetate	0.594
Heptane	1
Hexane	1
Methanol	1
Toluene	0.982
Average	**0.932**

TABLE IV: Validation and Test F1-Scores for different strategies approached in the 11 VOCs experiment

	F1-SCORE	
Strategy	**Val**	**Test(YOLO)**
CNN2D + Voting System [1]	0.974 [a]	0.727
CNN3D (BM 1) [1]	0.962	0.748
CNN3D (BM 1) – Stacking [1]	0.984	0.841
CNN3D (BM 1) – OvR [1]	-	0.745
CNN3D (BM 1) – Smaller Input [2]	0.950	-
CNN3D (BM 1) – Smaller Input Stacking [2]	0.979	0.816
CNN3D (BM 2) [2]	0.848	-
CNN2D + LSTM (BM 1) [2]	0.987	0.888
CNN2D + LSTM (BM 1) – Stacking [2]	0.991	0.910
CNN2D + LSTM (BM 2) [2]	0.977	0.908
CNN2D + LSTM (BM 2) – Stacking [2]	0.992	**0.932**

[1] Original Input Size: 148x148
[2] Smaller Input Size: 75x75
[a] Voting System not applied

of parameters was used to train the new network. On the validation set, the YOLO achieved a mAP of 0.87, a Recall around 0.90 and a Precision around 0.70. Even though these evaluation metrics are a bit lower than the same ones recorded on the YOLO network trained for the 11 VOCs experiment, as it is possible to observe on Figure 11, the detections are still very accurate. These are detections performed on a frame from the test set.

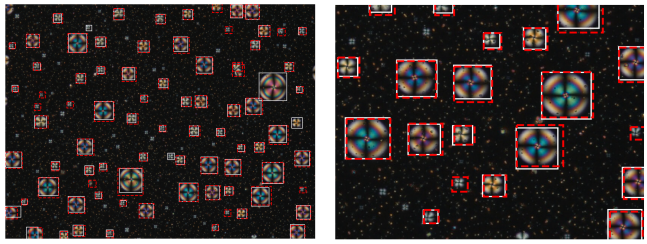

Fig. 11: Ground truth bounding boxes (White) vs YOLO detected bounding boxes (Red) - **Left**: Whole Frame **Right**: Frame Zoomed In

2) Concentration Regression: Again every droplet sequence had to be resized to a specific dimension in order to be fed to the different models. This time every droplet was resized to 100x100 pixels instead of being resized to the biggest dimension recorded in the dataset (237x237 pixels). Even though on the 11 VOcs experiment, the LSTM models outperformed, significantly, every CNN3D based models,

both variations were tested. A total of 4 different methods were tried, both CNN3D and LSTM networks with both CNN baseline models (BM1 and BM2). This time, instead of the Categorical Crossentropy, the Mean Squared Error loss function was used since this is now a regression problem. The optimizer was kept the same.

The target values for the models should be the concentrations of Acetone exposed to the gels. Instead, due to a small mistake, the targets were set to be the volumetric flow rates of Acetone. This is not problematic at all since both the concentrations and the flow rates are related in a linear way. Nevertheless, these values will still be referred to as concentrations for an easier understanding. The concentration/flow values selected for the 11 exposures were: 0.2, 0.28, 0.36, 0.44, 0.52, 0.6, 0.68, 0.76, 0.84, 0.92 and 1 (difference of 0.08 between consecutive exposures).

Out of the 4 tested models, curiously, the LSTM ones completely underachieved compared to the CNN3D ones which is the opposite of what happened in the 11 VOCs experiment. While the best CNN3D model (BM1) achieved a Mean Absolute Error (MAE) of 0.0993 for the droplet sequences on the test set, the best LSTM model (BM1) was only able to achieve a MAE of 0.1638. Since the difference in concentration between exposures was 0.08, both the achieved MAEs are definitely too far from what is considered acceptable. Figure 12 presents the predictions on the test set by the best CNN3D (BM1) model. There is a high fluctuation in predictions for droplet sequences belonging to the same exposure which shows that is very risky to guess the concentration based on a single droplet. Overall, there is a trend for the predictions to be below the expected values which happened in every tested model, both CNN3D and LSTM. In regards to the influence of the droplet size in the results, there wasn't a consistent trend throughout every experiment which is probably due to the very erroneous predictions.

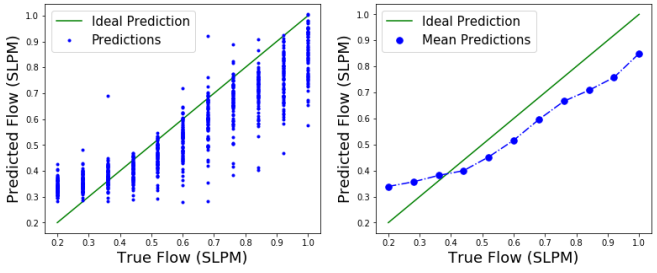

Fig. 12: CNN3D (Baseline Model 1) test set predictions

One of the decisions taken, coming into this problem, was not to perform any kind of cross validation in order to be able to experiment with different model architectures without the need of spending huge amounts of time on training. On the other hand, this implies a lesser level of confidence on the performance of the different models since only one evaluation is performed on each model. If by any circumstance, the data on the test set happens to have features that digress significantly from the data in the train and validation sets, the results might come out influenced by a high chance factor. Having extensive datasets is the most obvious solution to this problem since the data is more likely to cover a higher range of possibilities, decreasing the chance of already trained models facing new data with completely different characteristics. In this problem, as mentioned before, out of the 6 recorded videos, 5 were selected for the training stage of the models (4 for training and 1 for validation) and the remaining one for testing. This cannot be considered a extensive dataset, very far from it, so the bad results registered on the previous sections may be related to this fact and not to the inability of the different models to solve this task. To check whether this was or not the case, a new distribution of the videos across the different sets of data (train, validation and test) was randomly made, only making sure that the video that was previously selected for the test set was not selected again.

Two different networks were tested before for both the CNN3D and CNN2D + LSTM architectures (BM1 and BM2). The models that best performed on the validation set of each of these architectures were trained and tested again for these new sets of data, which are both coincidentally the implementations involving Baseline Model 2. Figure 13 presents the predictions of both these models on the test set. Again, the CNN3D model ended up performing better on the test set than the LSTM + CNN2D model, with these models achieving MAEs of 0.0490 and 0.0711. This represents a decrease of more than a half of the original values for both models. The main difference found on these results when compared to the original ones, besides the obvious increase in performance, is also the significant decrease in variance of the values predicted byboth models, specially the LSTM one. While before the predictions seemed to be a little bit all over the place, now they are definitely more compact and more centered on the true concentration values. This helped the mean predictions to follow more closely the true values.

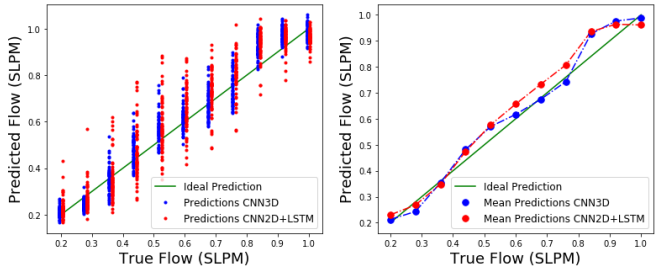

Fig. 13: CNN3D and CNN2D+LSTM new test set predictions

Now, with a more consistent set of predictions it was possible to get more conclusive results on the influence of the droplet size in the prediction error. Figure 14 presents these results. For both models, with the increase in droplet diameter it is registered a decrease in concentration error with an exception for the bigger droplets on the LSTM model, for which the reason is unknown.

Again, to wrap up the experiment, all results were put together. This time the results are presented in the form

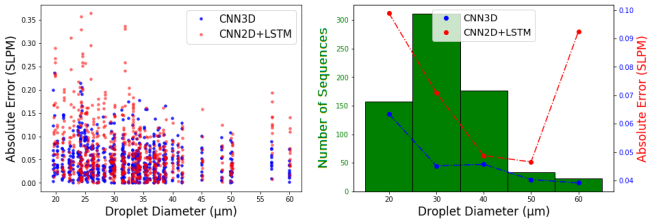

Fig. 14: CNN3D (Baseline Model 1) test set errors in function of the droplets diameters

TABLE V: Validation and Test MAE for different strategies approached in the Concetrations experiment

	MAE	
Strategy	**Validation**	**Test (YOLO)**
CNN3D (BM 1)	0.0546	0.0993
CNN3D (BM 2)	0.0418	0.1038
CNN2D + LSTM (BM1)	0.0553	0.1638
CNN2D + LSTM (BM2)	0.0514	0.1684
CNN3D (BM 2) [1]	0.0489	**0.0490**
CNN2D + LSTM (BM2) [1]	0.0605	0.0711

[1] Redistributed Dataset

of the mean absolute error and can be observed in Table V. As previously mentioned, for this experiment, the three dimensional CNNs were able to achieve significantly better results than the LSTMs on the test set. Even when it was thought that the dataset distribution could be affecting the final results, after the redistribution, the CNN3D network still managed to pull ahead and produce better results.

VI. Conclusions

In this work, the use of CNN based systems was explored in two different E-nose applications, one with the goal of detecting the substance exposed to the gel and the other with the goal of predicting the concentration of Acetone in the exposed gas. It was observed on both tasks that these systems were able to accomplish the objectives with satisfactory results.

In the 11 VOCs experiment, firstly the focus was around the CNN3D (Baseline Model 1) and trying to make it work by exploring many different strategies and parameter variations. Although the results overall were not bad, a few classes were constantly underachieving no matter what strategy was used so the focus had to be turned to another direction. A second deeper model (Baseline Model 2) was used and LSTM networks were tested. After a few tests it was soon realized that the results produced by the LSTM models were significantly better than all the ones achieved by the 3 dimensional CNNs. The reason behind this was probably the different way on which the LSTMs approach the temporal component of the data, which should be more suitable to this specific data/problem. By putting together several of these models under a stacking ensemble, close to perfection predictions on all classes were achieved, except for one (Ethylacetate) that missed about half of the times. Regarding the influence of the droplet diameters on the models' performance, it was acknowledged that the bigger droplets are capable of producing more accurate results on a more regular basis. Since relying on a single droplet to predict the VOC might be a bit risky, if all the detected droplets contributed to the final prediction, and by assigning a bigger weight to the bigger droplets, it seems that it should be possible to achieve very consistent results.

For the Concentrations experiment, curiously, the model that achieved better performance on the 11 VOCs experiment, was not the one that obtained the best ones here. For both the original and changed data distribution, the (Baseline Model 2) CNN3D was the model that achieved the lowest error values. In this experience, the impact that a different distribution in data, in a small dataset, can have in regards to the final results was noticed. While the predictions on the original test set were all over the place, mostly below the true values, on the second distribution the predictions were way more accurate and centered on the real values. Given this, it is very unlikely to get accurate results based on the predictions of singular droplets given the volatility presented on their predictions. Regarding the influence of the droplet size on this task, it is not 100% clear but it seems that the bigger ones are also able of giving more accurate predictions. This is mostly based on the results obtained on the last experience, with the changed sets, since the results on the original ones were disappointing and no assertive conclusions could have been made from them. So again, in order to get better results, bigger droplets should have a higher influence on the final prediction.

In terms of the YOLO networks used, they were always capable of identifying the droplets in images with great precision. During the testing phase, when some results were below expectations on the test set, they were compared with the ones obtained with the ground truth test sequences to check if the YOLO detections were having a big impact on the predictions. The ground truth results were always very similar to the ones obtained through YOLO detections so it can be concluded that this network was definitely a good choice to perform the droplet detection.

References

[1] C Sarafoleanu et al. "The importance of the olfactory sense in the human behavior and evolution". In: *Journal of medicine and life* 2.2 (2009), p. 196.

[2] Yoav Y Broza et al. "Hybrid volatolomics and disease detection". In: *Angewandte Chemie International Edition* 54.38 (2015), pp. 11036–11048.

[3] Morad K Nakhleh et al. "Diagnosis and classification of 17 diseases from 1404 subjects via pattern analysis of exhaled molecules". In: *ACS nano* 11.1 (2016), pp. 112–125.

[4] Gang Peng et al. "Diagnosing lung cancer in exhaled breath using gold nanoparticles". In: *Nature nanotechnology* 4.10 (2009), p. 669.

[5] Dedy Rahman Wijaya et al. "Development of mobile electronic nose for beef quality monitoring". In: *Procedia Computer Science* 124 (2017), pp. 728–735.

[6] Amy Loutfi et al. “Electronic noses for food quality: A review”. In: *Journal of Food Engineering* 144 (2015), pp. 103–111.

[7] LE Selena Sironi et al. “Use of an electronic nose for indoor air quality monitoring”. In: *CHEMICAL ENGINEERING* 40 (2014).

[8] Mahmoud Z Iskandarani. “A novel odor key technique for security applications using electronic nose system”. In: *American Journal of Applied Sciences* 7.8 (2010), p. 1118.

[9] Steve Lawrence et al. “Face recognition: A convolutional neural-network approach”. In: *IEEE transactions on neural networks* 8.1 (1997), pp. 98–113.

[10] Heng Wang and Cordelia Schmid. “Action recognition with improved trajectories”. In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 3551–3558.

[11] Fabio Alexandre Spanhol et al. “Breast cancer histopathological image classification using convolutional neural networks”. In: *2016 international joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 2560–2567.

[12] Danli Wu et al. “POP-CNN: Predicting Odor Pleasantness With Convolutional Neural Network”. In: *IEEE Sensors Journal* 19.23 (2019), pp. 11337–11345.

[13] Sai Xu et al. “Detection of type, blended ratio, and mixed ratio of pu’er tea by using electronic nose and visible/near infrared spectrometer”. In: *Sensors* 19.10 (2019), p. 2359.

[14] Akinori Hidaka and Takio Kurita. “Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks”. In: *Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications*. Vol. 2017. The ISCIE Symposium on Stochastic Systems Theory and Its Applications. 2017, pp. 160–167.

[15] Chigozie Nwankpa et al. “Activation functions: Comparison of trends in practice and research for deep learning”. In: *arXiv preprint arXiv:1811.03378* (2018).

[16] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[17] Cenk Temizel et al. “Production Forecasting in Shale Reservoirs Using LSTM Method in Deep Learning”. In: Unconventional Resources Technology Conference (URTEC). 2020.

[18] Jiang Guo. “Backpropagation through time”. In: *Unpubl. ms., Harbin Institute of Technology* 40 (2013).

[19] Corentin Tallec and Yann Ollivier. “Unbiasing truncated backpropagation through time”. In: *arXiv preprint arXiv:1705.08209* (2017).

[20] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[21] Joseph Redmon and Ali Farhadi. “YOLO9000: better, faster, stronger”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271.

[22] Carina Esteves et al. “Effect of film thickness in gelatin hybrid gels for artificial olfaction”. In: *Materials Today Bio* 1 (2019), p. 100002.

[23] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.