Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Automatic CNN-based Enhancement of 360° Video Experience with Multisensorial Effects

JOHN SEXTON[1], ANDERSON AUGUSTO SIMISCUKA[1], (Member, IEEE), KEVIN MCGUINNESS[1], and GABRIEL-MIRO MUNTEAN[1], (Senior Member, IEEE)

[1]School of Electronic Engineering, Dublin City University, Dublin, Ireland.

Corresponding author: Anderson Augusto Simiscuka (e-mail: anderson.simiscuka2@mail.dcu.ie).

This work was supported by the European Union's Horizon 2020 Research and Innovation program under Grant Agreement no. 870610 for the TRACTION project. The support of the Science Foundation Ireland (SFI) Research Centres Program via grants SFI/12/RC/2289_P2 (Insight) and SFI/16/SP/3804 (ENABLE), co-funded by the European Regional Development Fund, is also gratefully acknowledged. is gratefully acknowledged.

ABSTRACT High-resolution audio-visual virtual reality (VR) technologies currently offer satisfying experiences for both sight and hearing senses in the world of multimedia. However, the delivery of truly immersive experiences requires the incorporation of other senses such as touch and smell. Multisensorial effects are usually manually synchronized with videos and data is stored in companion files, which contain timestamps for these effects. This manual task becomes very complex for 360° videos, as the scenes triggering effects can occur in different viewpoints. The solution proposed in this paper aims to automatically add extra sensory information to immersive 360° videos. A novel scent prediction scheme using Convolutional Neural Networks (CNN) is proposed to perform scene predictions on 360° videos represented in the Equi-Angular Cubemap format in order to add scents relevant to the detected content. Digital signal processing is used to detect loud sounds in the video with a Root Mean Squared (RMS) function, which are then associated with haptic feedback. A prototype was developed, which outputs multisensorial stimuli by using an olfaction dispenser and a haptic mouse. The proposed solution has been tested and it achieved excellent results in terms of accuracy of scene detection, olfaction latency and correct execution of the relevant effects. Different CNN architectures, including AlexNet, ResNet18 and ResNet50, were also assessed comparatively, achieving a labeling accuracy of up to 72.67% for olfaction-enhanced media.

INDEX TERMS Multisensory, neural networks, three-dimensional visualization, immersive video.

I. INTRODUCTION

ADVANCES in visual-based media over the past few decades have created a massive market for cutting-edge design and innovation in the delivery of immersive experiences. However, increasing displays' pixel density or improving color contrast does not necessarily increase the perception of immersion in multimedia experiences [1].

Head-mounted virtual reality (VR) displays like the Oculus VR series or the HTC Vive are already popular among users. These displays fill the viewers' field of view completely with video content. Infrared cameras and sensors track the users' head movements, controlling the orientation of the display [2]. In conjunction with high-quality surround sound, VR technology is a big step towards an immersive and realistic experience. However, other senses such as touch and smell are also important aspects of immersive systems. Studies demonstrate that odors stimulate more memories related to human emotions than other stimulus types [3]. By adding more senses to existing media, a more stimulating and memorable experience may be created.

Olfaction devices blow air across smell-producing cartridges into the nose of the viewer to mimic scents in a video. This is an effective way to reproduce scents, and, typically, the scents are manually programmed to synchronize with the video [1], [4]. The same issue exists with haptics: vibration emitters have to be programmed to be executed at specific

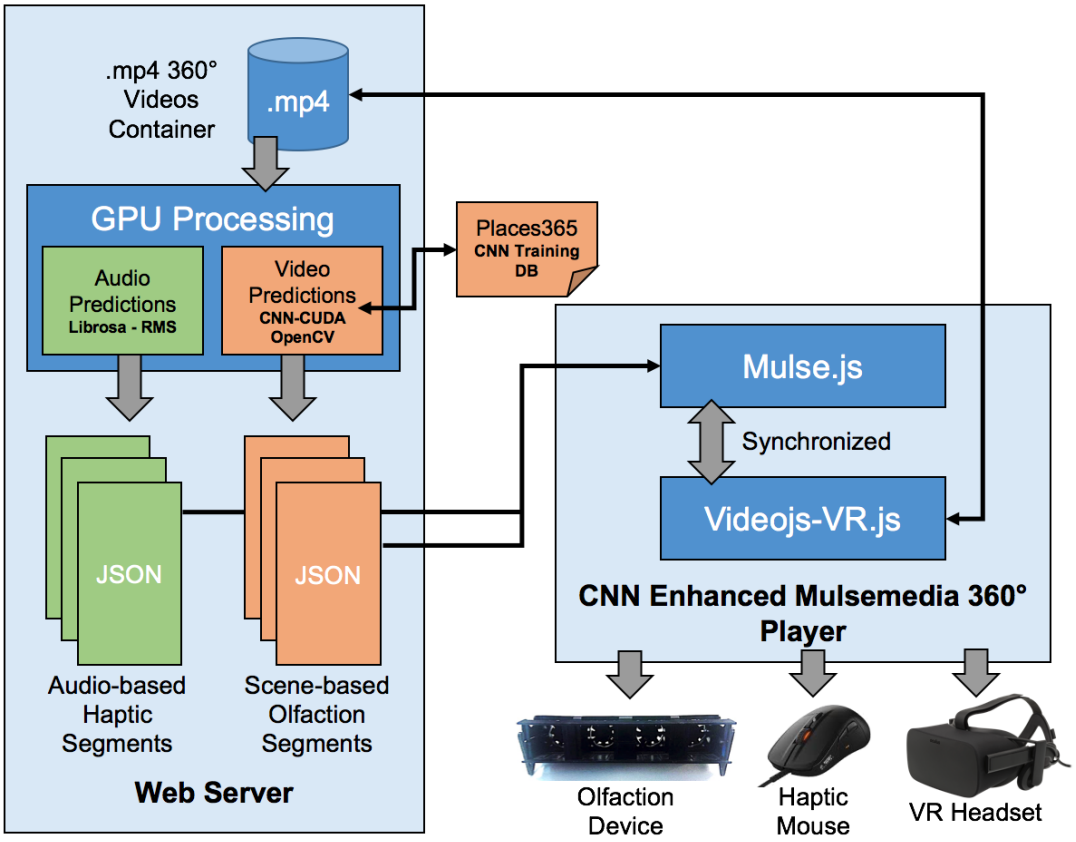


FIGURE 1. CNN-Enhanced Mulsemedia Architecture

points in the video [5], [6]. Synchronizing multiple sensorial media (mulsemedia) effects with videos is a time-consuming manual task, especially in lengthy videos. The duration of each effect must also be carefully adjusted. This synchronization task is even more complex for VR content [7], [8], which requires the sensorial effects to be activated based on the entire 360° field of view.

A possible solution to replace manual video tagging is to record scents and vibrations during production, but this increases the cost and does not take advantage of the wealth of existing immersive media items already available. A better solution involves addition of mulsemedia effects to existing VR videos; currently this is performed manually and is a very time-consuming process. The best solution should add multisensorial effects to existing media without human input and store the multisensorial-enhanced media data for playback. Scene recognition is required in this automatic multisensory media-enhancement process and a possible state-of-the-art scene recognition solution uses Convolutional Neural Networks (CNN) to accurately classify images.

Many challenges, however, need to be overcome while using CNN architectures for immersive content, as they are usually designed and trained to make predictions on 2D images [9], [10]. Pre-trained neural networks need to be adapted to make predictions on a 360° field of view. A suitable CNN architecture and a training model must also be selected based on the accuracy of scene classification. Label datasets currently do not support olfaction effects, which means they need to be customized to correlate scenes and scents. The amount of video frames that must be processed to generate adequate scents is also another parameter of interest, as it affects the CNN processing times. Regarding haptics, it is known that sound can sometimes be an analog for haptic feedback [11]. This requires a specific approach for audio processing and detection.

This paper aims to address these challenges, with the proposal of an innovative mulsemedia platform that performs scene recognition on immersive 360° content. Two algorithms are proposed, allowing for the automatic generation of scents and vibrations based on the audio-visual content. The algorithm for generating olfaction effects adapts different CNN architectures to support immersive content, using a tile-based scene recognition approach. An olfaction label dataset was also created, associating scents to the detected scenes. The proposed algorithm for generating haptic effects uses sound analysis. Digital signal processing is employed to sample the loudness of the signal and determine the peaks of loudness in the audio. If these peaks exceed a calibrated threshold, haptic feedback is produced. This is a novel approach to be integrated to a mulsemedia system, and increases users' sense of immersiveness.

A prototype, which includes a mulsemedia 360° player, was built for the testing of the solution, handling playback of the automatically generated mulsemedia content and in-

tegrating the four types of sensory media considered in the tests: audio, visual, haptics and olfaction. The architecture is presented in Fig.1, and will be described in details in the following sections. YouTube's Equi-Angular Cubemap 360° structure [12] is used in the player, as it composes VR video frames using a set of six 2D images. The video frames can be cropped into manageable sizes and processed by the CNNs.

Three CNN architectures were evaluated for the proposed solution: AlexNet, ResNet18 and ResNet50. The proposed algorithm for olfaction based on scene detection expands the field of view of these CNN architectures to support 360° content. When applying ResNet18 to the solution, a top-1 accuracy of 61.35% was achieved in scene classification. This accuracy corresponds to correct image detection obtained with the test dataset described in Section V.B. The proposed approach using ResNet18 also outperformed two other current works in terms of scene classification accuracy. Ultimately, testing the different setups led to a refined solution which provided an olfaction labeling accuracy of 72.67% on a test set consisting of 10,000 equi-angular cubemap frames from ten different 360° videos.

The remaining sections of this paper are organized as follows. Section II presents related works on the topics of mulsemedia, CNNs and immersive video processing. Section III discusses the solution design, including the architecture and the proposed algorithms for the generation of olfaction and haptic effects. Section IV describes the prototype developed for the testing of the proposed solution and Section V presents results related to olfaction latency, achieved accuracy and comparisons with other state-of-the-art approaches. Conclusions and directions for future work end this paper in Section VI.

II. RELATED WORK

A. MULTISENSORY MEDIA

The term mulsemedia has recently been used in the literature to describe media that engages three or more senses [5], [13]. There have been a number of works proposing mulsemedia solutions for 2D and 3D videos [4], [5]. Some mulsemedia solutions synchronize extrasensory data with multimedia content using the MPEG-DASH or MPEG-V standards [14].

The solution presented in [4] uses a smartphone-based headset and Arduino powered actuators to display 360° mulsemedia content, including scents and wind. The approach focuses on the subjective assessment of mulsemedia combined with 360° videos, and a prototype is used for the tests. Results indicate that the solution increases user's quality of experience.

The authors of [15] introduce an innovative server-based adaptive multisensorial media delivery solution to adjust the content to dynamically match the network capacity available. Researchers in [6] focus on the synchronization between multiple sensorial components, during mulsemedia content delivery, aiming to achieve high user perceived quality. The solution described in [14] introduces a client-based MPEG-DASH-based adaptation algorithm for multisensorial content distribution, which improves user quality of experience. The authors of [16] discuss the advantages and limitations of adaptive content delivery in MPEG-DASH, 2D, omnidirectional (360°) and multisensorial solutions. A solution that integrates scents and the video game Minecraft is proposed in [17], combining olfaction with audio-visual cues. The web-based mulsemedia player proposed in [5] performs 2D video mulsemedia playback over a network using MPEG-DASH to stream the video content with integrated scents and haptics.

Haptic feedback is an integral part of mulsemedia VR experiences. In VR gaming, the Oculus Rift provides haptic feedback to users, making game interactions more immersive. The work presented in [11] suggests that haptic effects can be linked to loud noises in audio-visual experiences. The authors also propose a haptic device called "HapticHead", which aims to increase realism in VR experiences by playing haptic events synchronized with loud noises and explosions. The Root Mean Squared (RMS) function is one of the most used solutions for detecting loud noises in sounds, as it finds the strength of the signal based on the amplitude [18]. Therefore, the automatic execution of sound-based haptic feedback can be based on an RMS threshold. Audio and haptics are also used in combination in a navigation tool for cyclists [19]. The vibrations, however, are executed together with every sound, without a threshold.

Unfortunately, these proposed solutions do not support neural networks, automatic detection of scenes and objects, and do not provide haptic effects based on audio signal features.

B. CONVOLUTIONAL NEURAL NETWORKS

Since the publication of Krizhevsky and Sutskever [9] in 2012, the potential of CNNs has been realized. Over the past decade CNNs revolutionized the fields of computer vision and machine learning. By training a network over a large dataset, the network recognizes patterns in images belonging to the same category. In the classification setting, the output of the final layer of the network forms a vector that has the same number of dimensions as there are possible labels/categories. This vector is softmax normalized to a probability distribution indicating the predicted category of the input. Common operators in convolutional networks are kernel convolution, max pooling, ReLU activations, softmax and batch normalization [20], [21], as outlined in the following subsections .

1) Kernel convolution

Kernel convolution is an operator that passes an $n \times n$ weighted kernel over an image, multiplying the corresponding weights with the array contents and summing the result. The result of each convolution is placed in the corresponding location in an output array. The weights in the kernel are initialized randomly. Using stochastic gradient descent combined with a backpropagation algorithm, which calculates the gradients of the parameters of the network with respect to a

loss function, the weights are iteratively adjusted to reduce the loss. This is called learning [21].

2) Max Pooling

Max pooling is used to gradually reduce the spatial dimension of the inputs to layers in CNNs. This reduces the number of parameters in subsequent fully connected layers and also decreases the computational complexity of subsequent convolutional layers. Max pooling passes a 2×2 window across an array and sets the value for that area of the array as the max value contained in the window [21], [22].

3) Rectified Linear Units

The Rectified Linear Unit (ReLU) is an activation function commonly used in CNNs. The function leaves positive values unchanged and sets negative values to zero, *i.e.*

$$f(x) = \max\{0, x\}.$$

This is usually applied to all values between convolutions to break linearity. Networks that use ReLU activations converge faster than other activation functions like $\tanh$ and the sigmoid function [9], [23]. They are also less affected by the vanishing gradient problem as larger inputs do not tend towards a horizontal asymptote as they do with $\tanh$ or the sigmoid [24].

4) Fully connected layers

Fully connected (or dense) layers are matrix multiplications. To apply such operations to the outputs of layers with spatial extent, the spatial extents must be removed in some way. Typically, this is done either by concatenation of spatial dimensions (as is done in the VGG and AlexNet networks) or by some form of global pooling (as in the ResNet networks). The last parametric layer in a CNN is usually a fully connected layer that projects the features to a space with the number of dimensions equal to the number of classes C.

5) Softmax

The output $\mathbf{z} \in \mathbb{R}^C$ of the final fully connected layer is generally not a categorical distribution over the classes. To convert it into a distribution, it must be normalized resulting in $\mathbf{z}_i \in [0, 1]$ and $\mathbf{z}^T\mathbf{1} = 1$. The softmax function is typically used for this:

$$\operatorname{softmax}(\mathbf{z})_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{C} e^{\mathbf{z}_j}}. \tag{1}$$

6) Batch normalization

The weights of deep neural networks are randomly initialized, typically by independent sampling from a zero-mean Gaussian with small variance. This is designed to keep the feature distribution at each layer approximately zero mean and unit variance. As the network gets trained, however, there can be a "covariate shift" in which the distribution of input features shifts away from a zero-mean Gaussian. This can substantially increase training times. Batch normalization, a technique in which the outputs of convolution layers are normalized achieving a zero mean and unit variance across the batch, is often used to mitigate this issue. Batch normalization also includes learnable weight and bias parameters to restore the representative power of the previous layer [25].

7) AlexNet and Residual Networks

Two of the main CNN architectures currently available are AlexNet and Residual Networks (ResNets). AlexNet [26] is one the first large scale CNN architectures with good performance on the ImageNet dataset classification. AlexNet outperformed previous non-deep learning-based models with significantly improved results. The AlexNet architecture contains eight layers: five convolutional layers, two fully connected hidden layers, and one fully connected output layer.

ResNets are more recent than AlexNet, achieving even better results. Input in ResNets not only passes through each layer of the architecture but is also able to bypass one or more layers to be summed directly with the output. This was developed by He *et al.* [27], for the creation of deeper networks. ResNet18 contains 17 convolutional layers, one average pooling, and a fully connected layer with an additional softmax layer. ResNet50 contains 49 convolutional layers with a fully connected layer at the end of the network. The problem with adding extra layers to classic CNNs is that the accuracy can saturate early due to gradient vanishing in the earlier layers. The ResNet architecture allows more layers to be added before this becomes an issue. This makes the ResNet a state-of-the-art architecture: it won the ILSVRC 2015 classification competition with a top-5 error rate of 3.57% on the ImageNet dataset.

C. IMMERSIVE VIDEO PROCESSING

Immersive videos in VR platforms, including the Google Cardboard and YouTube 360, are streamed in 360° video formats, such as the Equi-Angular Cubemap. These videos are normally encoded in high resolutions and are converted to the 3D space by a VR video player. The picture is split into a 3×2 grid which make up the faces of a cube as shown in the Fig. 2 [28]. Each frame is cropped and converted to a 3D image by a VR player. The player projects each face onto a cube rendered in the 3D space and interpolation smooths out the cube edges. The VideoJS-VR player supports this format and is employed in the proposed solution of this paper [29]. The cube faces of frames can be processed by a neural network individually, resulting in separate predictions. The Equi-Angular Cubemap format provides an even distribution of pixel density and homogeneous picture quality for VR headsets. The pixels in this format are more evenly spread than other common projections [12].

Neural network architectures vary depending on the problem that is being solved. CNNs have high accuracy on image recognition and have been applied to 2D video classification [30]–[32], saliency prediction [33] and depth estimation [34], but to the best of authors' knowledge not for detection and synchronization of multisensorial with audio-visual content.

The work proposed in [35] introduces a learnable graph neural network (GNN) for scene recognition trained on multiple datasets, including Places365, SUN397 and MIT67. The authors in [36] combine the use of CNN pre-trained models and a Layout Graph Network (LGN), enhancing the representation of spatial structures in the scene recognition process.

In the scheme proposed in this paper, different CNNs are tested to classify images after being trained on a dataset. A dataset in this context consists of a large set of images with labels that classify the images into certain categories. The two datasets initially considered for the tests were ImageNet and Places365. Both have previously been used to train CNNs with high levels of accuracy [9], [37]. Even though the ImageNet dataset has a broad set of labels, most do not fit correctly with scent categories. The Places365 dataset has 365 separate categories of scenery that can be related to multisensory feedback.

The current state-of-the-art accuracy on the ImageNet dataset of 300 million images with weak supervision is 88.5%, which was achieved by Facebook AI research in 2020 [38]. Evidently, image recognition on 2D images has reached high levels of accuracy with refinements to the standard CNN design. Several projects have extended this to 2D video [31], [32]. However, there has been a lack of research in employing CNNs for VR video.

III. SOLUTION DESIGN

The works discussed in the previous section have demonstrated that multisensorial stimuli, including olfaction and haptic effects, enhance user immersiveness during content delivery. However, the synchronization of mulsemedia effects with videos is a time-consuming, manually-performed task, especially difficult for long videos. This is because correct olfaction and haptic effects need to be released or executed at the right moments for appropriate durations. This synchronization task is even more complex in 360° videos, which contain relevant content for mulsemedia in a much larger visual field. Therefore, using CNNs to identify content in videos helps much by automatizing the generation of relevant olfaction effects, accelerating the process of creating new multisensorial experiences. A similar approach for automatic haptic feedback can be based on audio cues that trigger the effects. Loud noises, such as explosions, can be used to trigger vibrations via a controller or haptic mouse, increasing immersiveness, as users feel the impact with an additional sense.

The current state-of-the-art CNNs provide accurate and timely scene recognition for 2D videos. The 360° Equi-Angular Cubemap video format combines six 2D faces to form a 3D scene, as shown in Fig. 2. The novel solution described in this section employs this format when performing CNN-based scene recognition on the 360° content and analyzes each 2D cubemap face separately. This increases the accuracy of the recognition process, reducing the distortion normally present in 360° content and helps generate relevant

FIGURE 2. YouTube VR Equi-Angular Cubemap projection.

olfaction effects. Advances in digital audio signal processing also provide the audio cues needed for the generation of haptic effects in loud or impactful events of the video.

This section describes the proposed architecture and the two algorithms for the automatic generation of olfaction and haptic effects. Different CNN architectures, including AlexNet, ResNet18 and ResNet50, were assessed comparatively to evaluate the performance of the olfaction solution. A dataset containing customized labels for olfaction is proposed, associating scents to scenes. A prototype employs the proposed architecture and is evaluated in terms of scene classification and olfaction accuracy, and frame sampling rate, which is related to CNN processing times.

A. ARCHITECTURAL DESIGN

The proposed VR mulsemedia solution includes two main algorithms that automatically add olfaction and haptic effects to existing VR videos synchronized with the relevant audio-visual content. The multiple sensorial media generated by these algorithms is (dis)played to users by a mulsemedia player that synchronizes sensorial effects with 360° videos and employs a VR headset, an olfaction dispenser and a haptic device.

CNNs, in the context of olfaction prediction, are trained with the use of a labeled image dataset to distinguish features from unseen images. The pre-trained networks employed for the image recognition task accept 2D images as inputs. To allow VR video frames to be categorized using a 2D image pre-trained network, the proposed algorithm adapts the 360° imagery to be classified by the CNN. This algorithm is responsible for the pre-processing of the 3D images into 2D images, and also for combining the outputs of the different image fragments.

The proposed solution employs the YouTube's Equi-Angular Cubemap projection, which encodes VR video in 2D video formats with minimal distortion. Due to the minimal distortion, the proposed algorithm crops the cubemap image into the constituent faces and yields six 2D images suitable for input into a pre-trained network. Inputting these images individually generates six output vectors. Once the output has been combined into a single vector, the largest dimension of the vector indicates the output category. The output categories from the pre-trained network are not directly as-

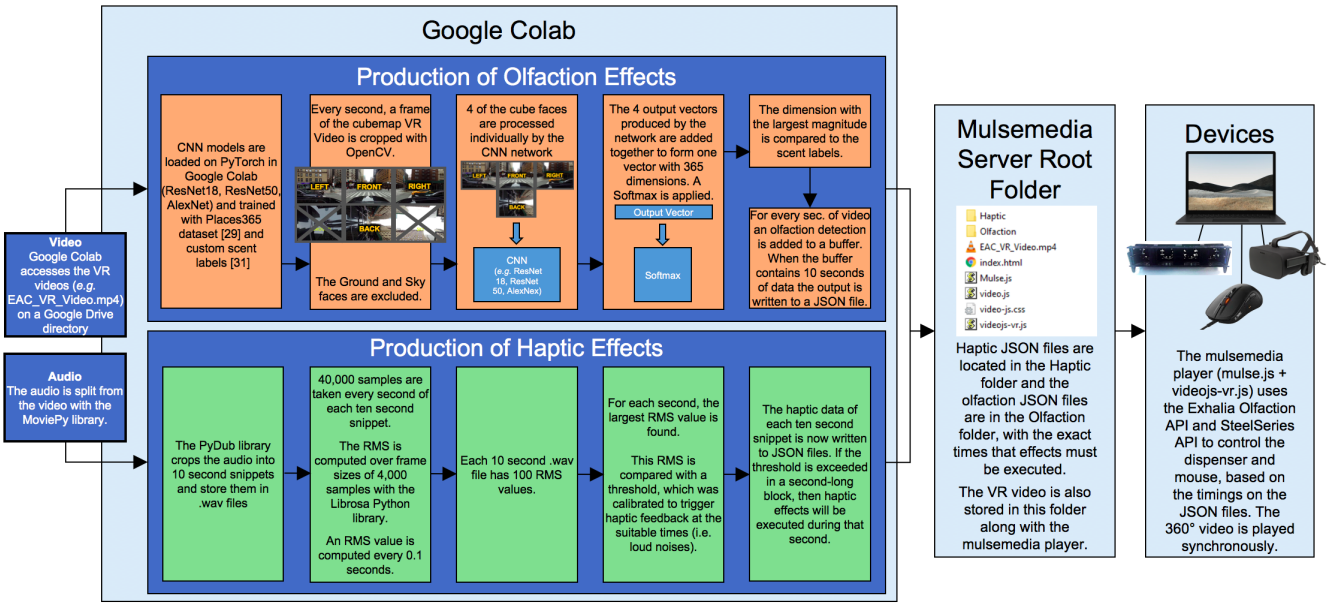


FIGURE 3. The process of automatically generating olfaction and haptic effects in 360° videos.

sociated with sensory information. The dataset Places365, however, contains labels for objects and scenery. These labels are customized with the relevant scents, which are associated to each scene.

Once the CNN is pre-trained, sampling of Equi-Angular Cubemap images from an input video must take place. The selection of the sample rate is described in Section V. The scene recognition process is executed in each of the sampled images, and once ten seconds of sensory data has been produced, the data is then written to a JSON file for playback following the sensory data storage convention created by Bi *et al.* [5].

The second algorithm produces haptic sensory data based on the audio input from the VR videos. Loud noises trigger the haptic feedback. The algorithm divides the audio in ten-second snippets, and then the digital signal is processed for the identification of the loudest noises. The algorithm detects the times and durations of audio bursts, based on a calibrated threshold, and associates haptic feedback to these periods.

The process of generating haptic and olfaction effects is summarized in Fig. 3, and the details of the proposed algorithms employed in this process are described in the following two subsections. In order to facilitate reproducibility, the source code for deployment of the solution is available on GitHub [39].

A prototype was also created for the testing of the approach. The prototype follows a client-server architecture and includes a web-based mulsemedia player that synchronizes the 360° videos with the olfaction and haptic effects described in JSON files, in a client-server architecture. The prototype uses an Oculus[1] Rift VR headset, an Inhalio[2] SBi4v2 olfaction dispenser and a SteelSeries[3] Rival 700 haptic mouse. The VR headset, olfaction dispenser and the haptic mouse are connected to a computer via HDMI and USB, as appropriate. The prototype implementation details are described in Section IV.

B. CNN-BASED ALGORITHM FOR THE GENERATION OF OLFACTION EFFECTS

The cloud-based programming platform Google Colab[4], which provides GPU support, was used for the deployment of the algorithm that generates olfaction effects, as shown in 3. The file `VideoPrediction.ipynb` contains the algorithm and can be executed on Google Colab.

This section details the process for the generation of olfaction effects. The machine learning algorithm for the generation of olfaction effects starts by processing MPEG-4 YouTube Equi-Angular Cubemap videos. Frames are sampled from the video and cropped into six smaller images (*i.e.* the faces of the cubemap, as seen in Fig. 2). Predictions on the separate faces are made by the pre-trained CNNs. Finally, the six prediction vectors are combined to form a single prediction for that 3D image.

[1]Oculus: https://www.oculus.com
[2]Inhalio: https://www.inhalio.com
[3]SteelSeries: https://www.steelseries.com
[4]Google Colab: https://colab.research.google.com

1) Using Pre-Trained CNN Models and Labels

The pre-trained CNN model (*e.g.* AlexNet, ResNet18, ResNet50) on Places365 is loaded into the PyTorch[5] open source machine learning library. PyTorch loads in the model architecture and the relevant weights directly from these files, without the need of a model class definition. Once the model states are loaded, the model is placed into evaluation mode. This turns off back propagation and training no longer occurs. After that, the network is set for making predictions. The device was set to run with CUDA, an API for Nvidia graphics chipsets which allows the model to run on GPUs, speeding up the processing time of the neural network. The CUDA API is also supported by Google Colab.

The images inputted into the CNN must be cropped and converted into a tensor before evaluation takes place. The network produces an output vector with 365 dimensions, each corresponding to the probability that the input image belongs in the corresponding category. To match the list of probabilities with a corresponding output label, an ordered list of the labels is loaded into the script. `Categories_places365.txt` is the text file containing the 365 labels which are outputted by the CNN. This file has been customized to associate a scent to each label and is available on GitHub [40].

Once a detection is made on a frame, the output from the algorithm is a number between 1 and 365, representing a label of the `Categories_places365.txt` file. Beside each label, a letter represents a scent which should be played when that scene is detected. For example, the row "/f/forest 120 o", indicates that 'forest' is the 120th label of the file and the letter "o" stands for the scent 'oak', which will be dispensed by the olfaction device when a scene with a forest is detected. A sample of the labels file and the scent encoding convention (in yellow) can be seen in Fig. 4.

```
Ocean = o
Oak = a
Candy = c
Chocolate = x
Diesel = d
None = n
```

```
/b/berth 55                    o
/b/biology_laboratory 56       o
/b/boardwalk 57                a
/b/boat_deck 58                o
/b/boathouse 59                o
/b/bookstore 60                a
/b/booth/indoor 61             a
/b/botanical_garden 62         a
/b/bow_window/indoor 63        n
/b/bowling_alley 64            c
/b/boxing_ring 65              n
```

FIGURE 4. The scent encoding (left) and a sample of categories_places365.txt with the scent labeling convention (right).

2) Frame Processing

Before CNN predictions, the equi-angular cubemap videos must be processed with an OpenCV-based Python script. The frame processing script is available in the file `ConvertVidsToImg.py`. Videos are converted into `cv2.VideoCapture` objects, which have the functionality to parse video frames at specified times in the video. Once a video is loaded, the frames per second and frame count of the video are used to calculate the exact duration of the video. The script takes a frame from the input video every second and send it to the prediction function, which determines the scene content and the correspondent scent.

3) The Prediction Function

Each equi-angular cubemap frame selected for processing must be cropped into its six constituent cube faces, so a prediction can be made on the separate faces. Frames are cropped into images of size $w \times h$ with the use of nested 'for' loops, which iterate through each face of the cubemap. For instance, if a frame has a resolution of 1800px horizontally and 1200px vertically, 6 smaller images will be created by cutting the main image at 600px and 1200px on the horizontal axis and 600px on the vertical axis. Each face is processed by the selected neural network and added to an accumulation vector that combines the prediction of all faces. Once all faces are predicted for that frame, a softmax is applied resulting on a probability distribution vector of the 365 categories. AlexNet, ResNet18 and ResNet50 were the CNN architectures tested for the prediction process.

During testing it was noted that using only four of the cube faces (*i.e.* the front, back, left and right faces) yielded an increase of 23.85% in olfaction accuracy instead of using all six faces (see Section V.C). It was also noted during testing that performing the softmax after combining the vectors together yielded better accuracy than performing the softmax on each vector individually before adding the vectors together (see Section V.D).

The top result from the vector returned from the prediction function determines the olfactory reading for that second of video, based on the labels presented in Fig. 4. The resulting scents are then added to the olfaction list.

4) Generating Olfaction Data

Once the olfaction list contains data for each ten seconds of video, the list is sent to the function *writeOlfJSON*. This function determines which scent is the most common during each ten-second period, and the starting time and the duration of the effect.

After that, the most common scents, their starting times and durations are written to JSON files, in the 'Olfaction' folder directory. These files are later read by the mulsemedia player (*i.e.* `mulse.js`) for playback.

C. ALGORITHM FOR THE GENERATION OF HAPTIC SENSORY DATA

The generation of haptic sensory data is based on the correlation of audio bursts and vibration, described in [11]. The algorithm, available in the file `VideoPrediction.ipynb`, uses digital signal processing for locating the key noises in ten-second periods. If a noise exceeds the calibrated noise level, the haptic device vibrates.

[5]PyTorch: https://www.pytorch.org

1) Parsing VR Audio into 10-Second-Long .wav Files

Audio must be processed separately from the video for the generation of haptic effects. The library MoviePy is used to parse audio tracks and store them in .wav files. The audio is then cropped into ten-second snippets using the Pydub library. The ten-second snippets are saved separately for digital signal processing.

2) Using RMS to Find Loudest Part of the Sound Signal

Once each ten-second snippet is ready to be processed, it is possible to predict where haptic feedback should be played, as the algorithm can now find the loudest noise in each snippet. This is done via the Root Mean Squared (RMS) function available in the Python Librosa library. The input audio files are sampled at 40,000 samples per second, resulting in 400,000 samples in each ten-second snippet of audio. The RMS function calculates the RMS of every 4,000 values and moves through the data in hop lengths of 4,000 samples. This returns an RMS value for every 0.1 seconds of audio. The following function is used to calculate the RMS to find the loudest noise every ten seconds:

$$\mathrm{RMS}_j = \sqrt{\sum_{i=1}^{4000} \frac{|x_{ij}|^2}{4000}}, \tag{2}$$

where x_{ij} is the i^{th} sample in the j^{th} frame.

3) Calibrating the Noise Threshold

The RMS array returns 100 values corresponding to a 0.1s period of audio. The maximum value in each ten-second period is compared with a threshold value of 0.51. This threshold value is calibrated to output haptic feedback for loud noises. This value was calibrated using sample audio signals, starting out lower and being increased until an adequate level of noise was reached.

4) Generating Haptic Data

The haptic effects starting times and durations are sent to the function *writeHapJSON*, which formats the data into JSON files. These JSON files are read by the mulsemedia player and are stored in the "Haptic" folder directory, ordered by their creation times.

IV. PROTOTYPE IMPLEMENTATION

Once the extra sensory data has been produced and stored in JSON files, the next step is to produce synchronized playback of the video, audio, haptic and olfaction data. The setup of the hardware can be seen in the Fig. 5. Audio and video are played through the Oculus headset, while scents are produced by the Inhalio olfaction dispenser placed in front of the user. The dispenser contains four scent cartridges and the SteelSeries haptic mouse provides haptic feedback. The VR videos and the mulsemedia content are hosted on a web server that also contains the mulsemedia player. The source code is available in [39]. The playback of mulsemedia content was tested on a client PC running the Firefox browser,

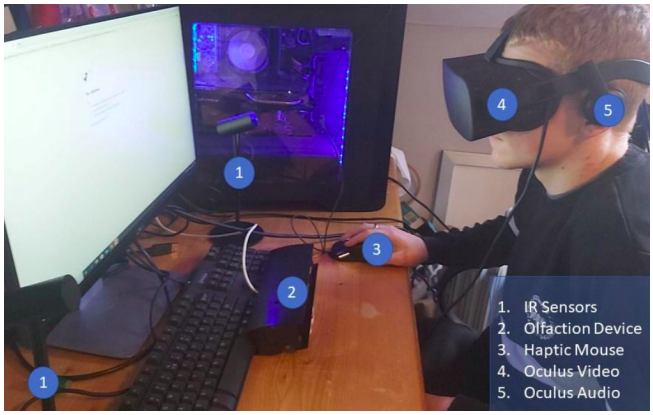


FIGURE 5. A prototype setup of the mulsemedia playback environment.

with 16GB RAM, Intel Core i7 processor and Windows 10. This section details the details for implementing the prototype, including the mulsemedia player developed for testing the solution.

A. SERVER-SIDE CONFIGURATION

As described in Section III, the 'Haptic' and 'Olfaction' directories both contain JSON files with their respective types of multisensory data. These files are requested by the client PC running the web-based mulsemedia player available in the `mulse.js` JavaScript file and its companion HTML file. The HTML page of the web application contains the embedded libraries for VR video playback: `video.js`, `video-js.css` and `videojs-vr.js`, which are contained in the primary directory. The VR video `RedwoodsWalkAmongGiants.mp4` is the 360° video used for testing the prototype.

All JavaScript files on the server side are included in the web application web page. JQuery is also included as it is used to send HTTP GET requests to the APIs for haptics and olfaction control. The 'video' HTML tag defines the video source and provides the functions for the actions to take when the video plays, pauses or ends.

VideoJS-VR converts the standard video tag to a VR player with the YouTube Equi-Angular Cubemap set as the projection type of the VR player. VideoJS-VR automatically detects VR head-mounted displays using the WebVR API built into the Firefox web browser.

B. JAVASCRIPT MULSEMEDIA PLAYER IMPLEMENTATION

The mulsemedia player synchronizes the mulsemedia effects with the VR video as soon as the video starts to play.

The function *JSONfunc* requests the olfaction JSON files from the server and parses the data from each file. The duration and fan number are passed to the *play_olfaction* function, which interacts with the Inhalio scent dispenser API via HTTP requests. Provided the video is still playing, the function is recursively called every ten seconds. The process restarts and it requests the subsequent JSON file.

TABLE 1. Table of all readings taken during the olfaction experiment (in seconds).

1.901	2.013	2.088	2.312	1.747	1.873	2.011	1.607	2.033	1.722
1.821	2.137	1.902	2.102	2.231	1.487	1.605	1.816	1.799	2.738
2.314	2.201	1.653	1.891	1.605	1.400	1.908	1.635	1.893	2.385
2.801	1.802	1.893	1.791	1.608	1.462	1.517	1.713	1.491	1.576
2.523	1.921	1.972	1.764	1.533	2.356	2.063	3.200	1.959	1.875
2.741	2.203	2.024	1.678	2.098	1.685	1.613	1.868	1.480	1.920
1.801	2.304	2.118	2.197	1.586	1.683	1.951	1.819	1.584	1.873
1.901	1.696	1.991	1.898	1.998	1.783	1.665	2.230	2.121	1.989
1.779	1.702	2.301	1.881	1.638	2.151	1.833	2.348	1.533	1.453
1.701	2.124	2.101	1.674	1.917	2.233	2.360	1.462	2.312	1.717

The Inhalio olfation devices are connected to clients via USB. The devices are controlled via a dynamic linked library that provides multiple functions to control the olfaction fans. One of these functions takes an input string containing the fan number, duration, and fan intensity, which are needed for the execution of the olfaction content.

The *play_olfaction* function uses string concatenation to parse the fan number into a *"SCENT_x"* string. This string along with the duration of the effect is required for the interaction with the olfaction dispenser API. The API consists of a local Java servlet, which controls the hardware. Once the string has been created, it is sent to the port 4000 on the localhost address using JQuery. JQuery sends the string via a HTTP GET request and once the API receives the request it generates the desired scent by activating the selected fan containing the scent cartridge.

The USB-based SteelSeries haptic mouse produces haptic feedback via an API provided by the manufacturer. The API is based on HTTP requests, with functions used to trigger the mouse vibrations. The functions included are *send_game_event*, *bind_game_event* and *do_post*. During video playback, the *JSONfuncHaptic* function requests and parses the JSON files containing the times that haptic effects should be executed. After that, the function calls the mouse API with POST requests at the times retrieved from the JSON file.

V. RESULTS AND DISCUSSION

The prototype was employed in a number of tests that demonstrate the feasibility of the solution for automatic generation of mulsemedia effects for 360° content. First, an investigation of the lingering effect of the scents is provided. The results of this study were used to define how often frames of the videos in the test dataset should be classified by the CNNs. Next, another experiment focused on determining which faces of the cubemap videos must be processed for the most accurate results. The effects of the softmax layer on the 360° content was also examined, as well as the performance of different CNN architectures. Finally, state-of-the-art approaches were also compared against the proposed solution, and the accuracy of classifying the cubemap faces separately versus classifying them as one image was measured.

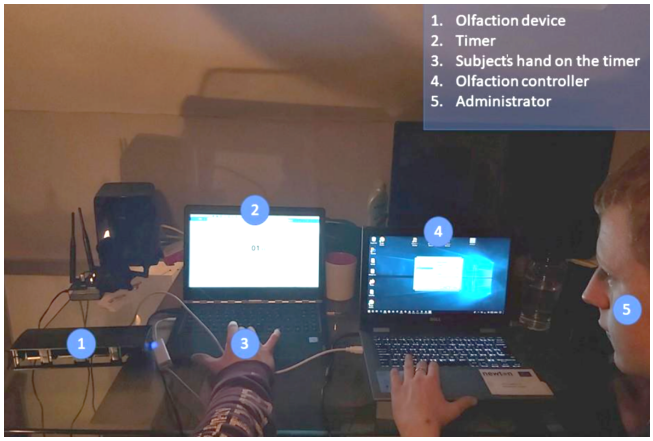


FIGURE 6. Setup of the olfaction sampling rate experiment.

A. OLFACTION LATENCY TESTING

Human vision and audio senses have a very high sample rate. Video and audio capture is only effective with very high sample rates (*i.e.* standards rates are 24Hz-30Hz for video or 44.1 kHz for sound [41], [42]). It was initially assumed that a high sample rate for olfaction should be used, which means that every video frame would be processed by the CNN. The high sample rate, however, resulted in poor performance of the olfaction algorithm, which took long periods of time to execute even in short videos. Therefore, a minimum adequate sample rate needs to be determined, considering how quickly a person can detect a transition in scents dispensed by the olfaction device.

To measure the average time a change in scent is detected, a simple experiment was devised. In this experiment, viewers sit fifty centimeters away from the olfaction device, blindfolded, and wearing ear protectors to reduce the effect of any external factors. The scent dispenser is loaded with two different scents (*e.g.* diesel and tea tree). A scent is dispensed for 5 seconds, and immediately afterward another scent is dispensed. Once the second scent starts getting dispensed, a timer is started. When the viewer notices a difference in smell, the timer is stopped. Fig. 6 presents the experiment setup.

A hundred readings were taken for the experiment, as detailed in Table 1, and the results showed an average of 1.924 seconds for olfaction transition. These results aided the decision for the sampling rate of CNN-based scene detections

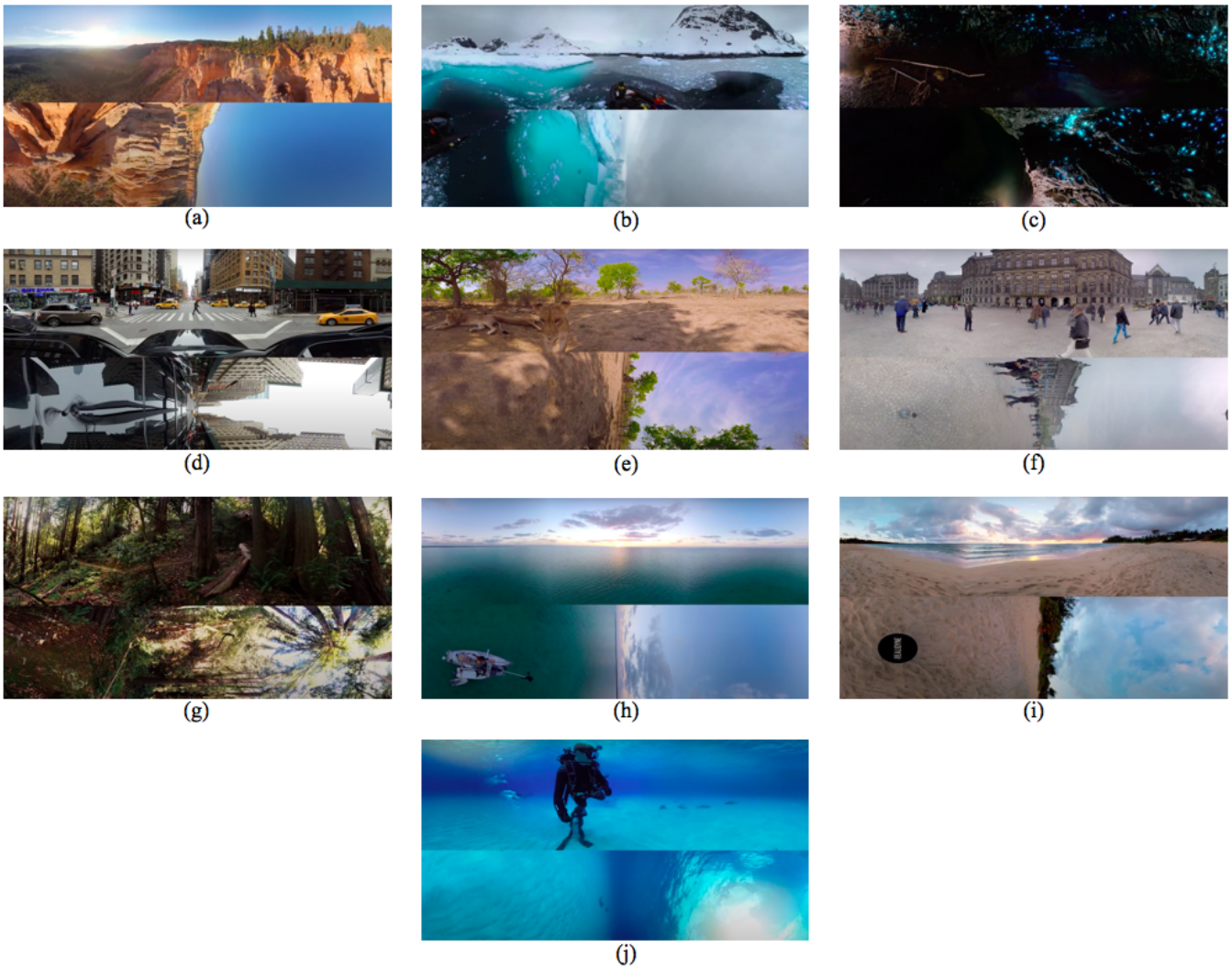


FIGURE 7. Screenshots of the videos used in the testing process.

for olfaction predictions. A sampling rate of one frame per second was decided upon, meaning that one cubemap frame is processed by the neural network per second of video. This is approximately twice the average scent change frequency of 1.924s detected in the experiment.

B. TEST DATASET

To test the pre-trained CNN network, a dataset of equiangular cubemap images was created from ten separate videos downloaded from YouTube, as seen in Fig. 7 [28], [43]–[51]. Each video was parsed into a thousand frames, creating a dataset of 10,000 images.

The selection of the frames in each video was spread equally through the video duration (*i.e.* the number of frames in the video ÷ 1000 = sampling interval. One frame is selected per sampling interval). Each image of the dataset contains all six faces of the 360° view, and they receive a label, manually annotated. These labels are then compared against the label returned by the automatic detection, so the accuracy of detection can be determined.

C. INTRODUCTION OF NOISE BY CERTAIN FACES OF THE CUBEMAP

While developing the olfaction generation algorithm, it was noted that the algorithm had a high probability of returning labels such as “sky” and “catacomb” during video predictions. These would sometimes appear in the top prediction erroneously, resulting in an incorrect output for olfactory

TABLE 2. Effect of the number of faces of the equiangular cubemap used in prediction accuracy.

Architecture	No. of faces used for prediction	Top-1 Accuracy	Top-5 Accuracy	Olfaction Accuracy
ResNet18	4	61.35%	86.67%	72.67%
ResNet18	6	36.37%	85.18%	48.82%

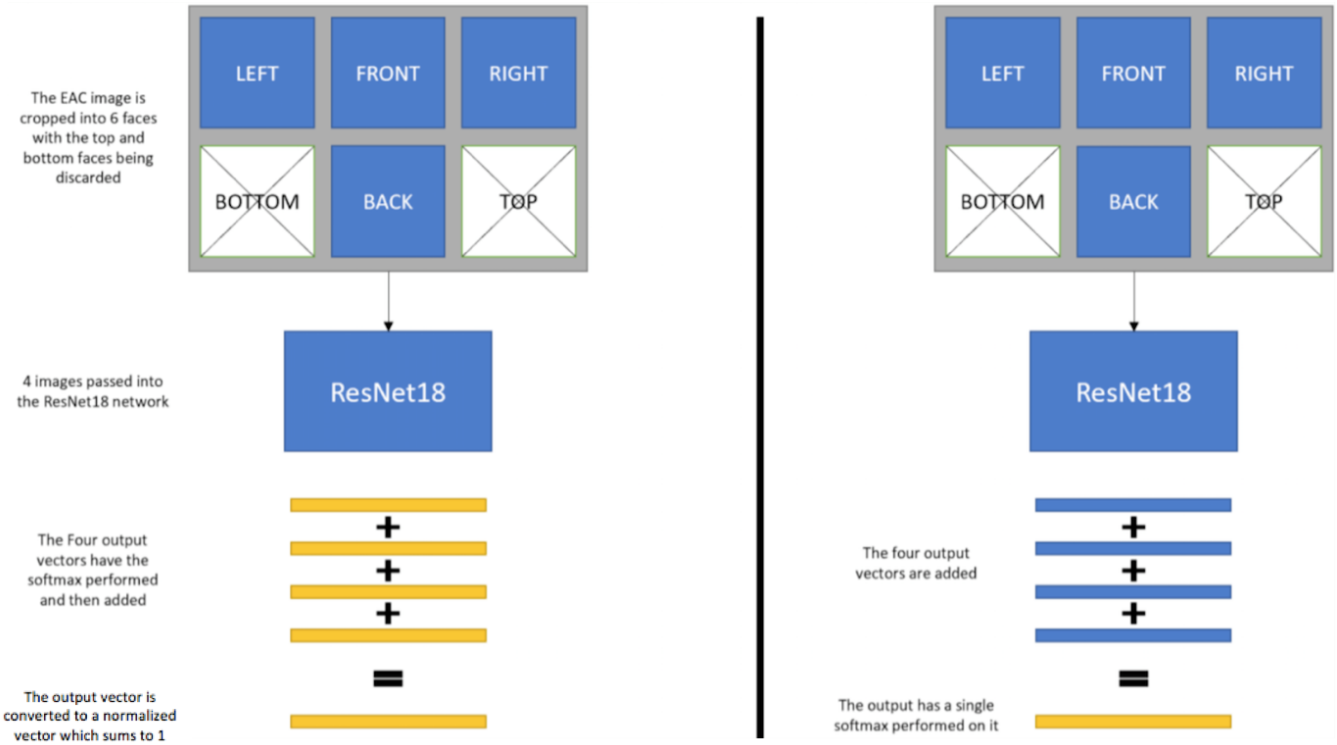


FIGURE 8. The four softmaxes setup (left) and the one softmax setup (right).

content. To discover what was causing these errors, the outputs for each face of the cubemap were examined individually. It was found that the upward face of the cubemap was often being classified with the "sky" label and the downward face receiving the label "catacomb", most likely due to the color of the ground.

It was hypothesized that making predictions on the top and bottom faces of cubemaps was adding levels of noise into the prediction process, decreasing accuracy. To investigate whether this hypothesis is accurate, the dataset of equi-angular cubemap images described in Section V.B was used for the tests. Each of the ten videos was parsed into a thousand frames, resulting on a dataset of 10,000 images. The accuracy of using all six faces for predictions versus using the four horizontal faces is tested using this dataset. Results are displayed in Table 2. The tests were carried out using the ResNet18 architecture with a single softmax applied after all predictions have been combined.

The results from this test showed a 23.85% increase in correct scent prediction while only using the four horizontal faces, with a top-1 accuracy of 61.35% for scene recognition and 72.67% accuracy in scent prediction accuracy. The top and bottom faces affected the top-1 accuracy significantly (a decrease of 24.98%), while the top-5 accuracy was only slightly reduced (a decrease of 1.49%). Scent predictions are associated with the top output of the algorithm, and for this reason, a high top-1 accuracy is a better metric than top-5, in relation to scent prediction. It is concluded from this experiment that using the four horizontal cubemap faces yields a better accuracy than using all six faces. Further tests use this setup as a result.

D. SOFTMAX AND VECTOR ADDITION

When combining the predictions of separate faces, the resulting vectors are added together to form a probability distribution. Output vectors are converted to a vector probability distribution by using the softmax function. Initially, when using the pre-trained network, a softmax was applied before the four vectors were added together resulting on an output vector with a magnitude of four. The output vector can be divided by four to become a normalized 365-dimensional vector that sums to 1. Another possible configuration, which can potentially offer a better separation between outputs, is to add the four vectors and perform a single softmax on the output vector. The layouts of the two setups are illustrated in Fig. 8.

The results from Table 3 contain the accuracy comparison of the four softmaxes before addition versus the one softmax after addition. The accuracy is measured based on the dataset described in Section V.B.

The results show a substantially higher accuracy when performing a single softmax as the final operation. The final softmax setup produces an olfaction accuracy 12.64% higher than using four softmaxes before addition. The single softmax configuration is employed due to the higher yielded accuracy indicated in the tests.

TABLE 3. Performance testing of the two softmax setups.

Architecture	No. Softmaxes Performed	Top-1 Accuracy	Top-5 Accuracy	Olfaction Accuracy
ResNet18	1	61.35%	86.67%	72.67%
ResNet18	4	48.27%	82.72%	60.03%

TABLE 4. Accuracy testing of different CNN architectures.

Architecture	Top-1 Accuracy	Top-5 Accuracy	Olfaction Accuracy
ResNet18	61.35%	86.67%	72.67%
ResNet50	47.97%	86.53%	61.11%
AlexNet	56.65%	84.71%	71.52%

TABLE 5. Comparison of scene recognition accuracy with state-of-the-art networks trained with the Places365 dataset.

Method	Top-1 Accuracy	Top-5 Accuracy
SE-GNN [35]	55.21%	80.42%
LGN [36]	56.50%	86.24%
Ours (ResNet18)	61.35%	86.67%

TABLE 6. Effects of processing cubemap faces separately and combined.

Setup	Top-1 Accuracy	Top-5 Accuracy	Olfaction Accuracy
Developed Algorithm with cropping method	61.35%	86.67%	72.67%
ResNet18 without cropping method	27.85%	58.53%	54.38%

E. ACCURACY COMPARISON OF CNN ARCHITECTURES

The accuracy of three different CNN architectures employed in the image detection solution was compared. The AlexNet, ResNet18 and ResNet50 architectures were trained on the Places365 dataset. All three architecture configurations were tested using the dataset presented in Section V.B and the results are shown in Table 4.

The results from the experiment presented in Table 4 indicate that the accuracy obtained by ResNet18 is higher than AlexNet and ResNet50 by a significant margin. ResNet18 achieved a top-1 accuracy 13.38% higher than ResNet50 and 4.7% higher than AlexNet.

Even though ResNet50 has been reported to achieve a higher accuracy than ResNet18 when trained on the ImageNet dataset [52], the authors in [53] also verified that ResNet18 can outperform ResNet50, when trained on the Places365 dataset. The olfaction accuracy, which is dependent on the scene classification accuracy is also the highest when employing ResNet18.

Based on these results, the ResNet18 network is the recommended architecture to be used in the implementation of the algorithm for the generation of olfaction effects for 360° content. The higher accuracy in scene detection provided by ResNet18 results in more accurate olfaction effects being dispensed to users.

The proposed scene recognition approach using ResNet18 also performed well in comparison to two other recent networks [35], [36] trained in the Places365, in terms of top-1 (an increase of 6.14% and 4.85%, respectively) and top-5 (an increase of 6.25% and 0.43%, respectively) scene recognition accuracy. The results are presented in Table 5. These networks were also trained with the Places365 dataset to perform scene classification. The gain in our approach comes from the fact that multiple faces of the cubemap are processed, increasing the accuracy of the classification process.

F. PROCESSING CUBEMAP FACES SEPARATELY OR COMBINED

An additional test was performed to measure the increase in accuracy achieved by the designed olfaction algorithm (*i.e.* cropping the cubemap faces of the selected frames and processing them separately) versus classifying the entire equi-angular cubemap frames with the ResNet18 network using resize and center crop (*i.e.* keeping all cubemap faces in one image and resizing it to fit the ResNet18 input size). The results can be seen in Table 6.

The results indicate a large increase in accuracy while using the algorithm, since cropping the faces of the cubemap reduces frame distortion. Combining the outputs from the cropped frames together yields an increase in scene detection accuracy from 27.85% to 61.35%, when compared with processing entire video frames directly into the network. This represents a 33.5% increase for top-1 accuracy and an 18.29% increase for olfaction accuracy.

VI. CONCLUSION AND FUTURE WORK

This paper presented the design, implementation and testing of an innovative CNN-based mulsemedia solution, which automates the process of adding sensorial effects to immersive videos. Two algorithms were proposed to generate olfaction effects using scene classification techniques on 360° videos and haptic effects based on audio cues. Different CNN architectures were evaluated in the scene classification process: AlexNet, ResNet18 and ResNet50. These architectures have been adapted to process 360° content, handling the different areas of equi-angular cubemap videos separately, providing an accurate final recognition. A suitable approach for processing the immersive videos was achieved after experiments were performed to test the solution in terms of frame sampling, accuracy and cubemap faces selection. Olfaction effects were more accurate with scene recognition being performed on the four horizontal faces of the cubemap frame.

A label dataset for olfaction effects was also proposed, correlating different scents to a variety of scenes. A prototype containing a VR headset, an olfation dispenser and a haptic mouse was built, combining audio, immersive video, olfaction, and haptic feedback. The prototype also contains

a 360° player that synchronizes the automatically generated mulsemedia effects with videos.

Tests indicated that the accuracy of the proposed approach for olfaction correctly generated scents 72.67% of the time, when ResNet18 was employed. Moreover, a top-1 accuracy of 61.35% for scene recognition in 360° videos was achieved, proving the feasibility of using CNNs and ResNet18 for image detection in immersive videos. Most pioneering detection algorithms have notably been considered successful while yielding similar results. This was also demonstrated by the approach outperforming two other state-of-the-art solutions.

Future work directions include optimizing the use of machine learning algorithms for real-time predictions, as this is computationally expensive. Other neural networks and training models that detect actions and human behavior can be considered for the generation of a larger variety of mulsemedia effects. The sensorial effects can also be applied to guiding users experiencing 3D content, providing additional directional cues. Finally, standardized libraries and datasets exclusive to mulsemedia experiences could also be proposed.

REFERENCES

[1] A. Covaci, L. Zou, I. Tal, G. M. Muntean, and G. Ghinea, "Is multimedia multisensorial? - A review of mulsemedia systems," ACM Computing Surveys, vol. 51, no. 5, Aug. 2018.

[2] D. Heaney, "How virtual reality positional tracking works," 2019. [Online]. Available: https://venturebeat.com/2019/05/05/how-virtual-reality-positional-tracking-works/

[3] R. S. Herz, "Are Odors the Best Cues to Memory? A Cross-Modal Comparison of Associative Memory Stimuli," Annals of the New York Academy of Sciences, vol. 855, no. 1, pp. 670–674, Nov. 1998.

[4] I. S. Comsa, E. B. Saleme, A. Covaci, G. M. Assres, R. Trestian, C. A. Santos, and G. Ghinea, "Do I Smell Coffee? The Tale of a 360° Mulsemedia Experience," IEEE MultiMedia, vol. 27, no. 1, pp. 27–36, Jan. 2020.

[5] T. Bi, A. Pichon, L. Zou, S. Chen, G. Ghinea, and G.-M. Muntean, "A DASH-based Mulsemedia Adaptive Delivery Solution," in Proc. International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE), Jun. 2018, pp. 1–6.

[6] Z. Yuan, T. Bi, G.-M. Muntean, and G. Ghinea, "Perceived Synchronization of Mulsemedia Services," IEEE Transactions on Multimedia, vol. 17, no. 7, pp. 957–966, May 2015.

[7] A. A. Simiscuka and G.-M. Muntean, "Synchronisation Between Real and Virtual-World Devices in a VR-IoT Environment," in Proc. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Jun. 2018, pp. 1–5.

[8] A. A. Simiscuka, T. M. Markande, and G.-M. Muntean, "Real-Virtual World Device Synchronization in a Cloud-Enabled Social Virtual Reality IoT Network," IEEE Access, vol. 7, pp. 106 588–106 599, Aug. 2019.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, Jun. 2017.

[10] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition Using Places Database," in Proc. International Conference on Neural Information Processing Systems (NIPS), Dec. 2014, pp. 487–495.

[11] O. B. Kaul and M. Rohs, "Wearable head-mounted 3D tactile display application scenarios," in Proc. International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI). New York, NY, USA: Association for Computing Machinery, Inc, Sep. 2016, pp. 1163–1167.

[12] C. Brown, "Bringing Pixels Front and Center in VR Video," 2017. [Online]. Available: https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/

[13] I. Tal, L. Zou, A. Covaci, E. Ibarrola, M. Bratu, G. Ghinea, and G.-M. Muntean, "Mulsemedia in Telecommunication and Networking Education: A Novel Teaching Approach that Improves the Learning Process," IEEE Communications Magazine, vol. 57, no. 11, pp. 60–66, Nov. 2019.

[14] L. Zou, T. Bi, and G.-M. Muntean, "A DASH-Based Adaptive Multiple Sensorial Content Delivery Solution for Improved User Quality of Experience," IEEE Access, vol. 7, pp. 89 172–89 187, Jul. 2019.

[15] Z. Yuan, G. Ghinea, and G.-M. Muntean, "Beyond Multimedia Adaptation: Quality of Experience-Aware Multi-Sensorial Media Delivery," IEEE Transactions on Multimedia, vol. 17, no. 1, pp. 104–117, Jan. 2015.

[16] A. Yaqoob, T. Bi, and G.-M. Muntean, "A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities," IEEE Communications Surveys Tutorials, vol. 22, no. 4, pp. 2801–2838, Jul. 2020.

[17] S. Alraddadi, F. Alqurashi, G. Tsaramirsis, A. Al Luhaybi, and S. M. Buhari, "Aroma Release of Olfactory Displays Based on Audio-Visual Content," Applied Sciences, vol. 9, no. 22, Nov. 2019.

[18] S. K. Shah, Z. Tariq, and Y. Lee, "IoT based Urban Noise Monitoring in Deep Learning using Historical Reports," in Proc. IEEE International Conference on Big Data (Big Data), Dec. 2019, pp. 4179–4184.

[19] A. I. Giesa, "Navigating Through Haptics and Sound: A Non-visual Navigation System to Enhance Urban Bicycling," Lecture Notes in Computer Science, vol. 12201, pp. 640–652, Jul. 2020.

[20] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification," Frontiers in Neuroscience, vol. 11, pp. 1–12, Dec. 2017.

[21] W. Wang and Y. Yang, "Development of Convolutional Neural Network and Its Application in Image Classification: A Survey," Optical Engineering, vol. 58, no. 04, pp. 1–19, Apr. 2019.

[22] J. Lin, L. Ma, and J. Cui, "A Frequency-Domain Convolutional Neural Network Architecture Based on the Frequency-Domain Randomized Offset Rectified Linear Unit and Frequency-Domain Chunk Max Pooling Method," IEEE Access, vol. 8, pp. 98 126–98 155, May 2020.

[23] J. Si, S. L. Harris, and E. Yfantis, "A Dynamic ReLU on Neural Network," in Proc. IEEE Dallas Circuits and Systems Conference (DCAS), Nov. 2018, pp. 1–6.

[24] X. Hu, P. Niu, J. Wang, and X. Zhang, "A Dynamic Rectified Linear Activation Units," IEEE Access, vol. 7, pp. 180 409–180 416, Dec. 2019.

[25] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, "Batch normalized recurrent neural networks," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, May 2016, pp. 2657–2661.

[26] Y. Guo, Z. Pang, J. Du, F. Jiang, and Q. Hu, "An Improved AlexNet for Power Edge Transmission Line Anomaly Detection," IEEE Access, vol. 8, pp. 97 830–97 838, May 2020.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 770–778.

[28] "New York City 8K - VR 360 Drive," 2018. [Online]. Available: https://www.youtube.com/watch?v=2Lq86MKesG4

[29] T. Deppisch, N. Meyer-Kahlen, B. Hofer, T. Łatka, and T. Żernicki, "HOAST: A Higher-Order Ambisonics Streaming Platform," in Proc. Audio Engineering Society Convention 148, May 2020, pp. 1–5.

[30] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-Scale Video Classification with Convolutional Neural Networks," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2014, pp. 1725–1732.

[31] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015, pp. 4694–4702.

[32] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting Image-trained CNN Architectures for Unconstrained Video Classification," in Proc. British Machine Vision Conference (BMVC). British Machine Vision Association and Society for Pattern Recognition, Mar. 2015, pp. 1–9.

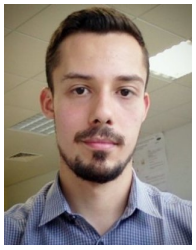
[33] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun, "Cube Padding for Weakly-Supervised Saliency Prediction in 360° Videos," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2018, pp. 1420–1429.

[34] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "BiFuse: Monocular 360° Depth Estimation via Bi-Projection Fusion," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020, pp. 459–468.

[35] J. Qiu, Y. Yang, X. Wang, and D. Tao, "Scene Essence," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2021, pp. 8322–8333.

[36] G. Chen, X. Song, H. Zeng, and S. Jiang, "Scene Recognition With Prototype-Agnostic Scene Layout," IEEE Transactions on Image Processing, vol. 29, pp. 5877–5888, Apr. 2020.
[37] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.
[38] H. Touvron, A. Vedaldi, M. Douze, and H. Jegou, "Fixing the Train-Test Resolution Discrepancy," in Proc. Advances in Neural Information Processing Systems (NeurIPS), Dec. 2019, pp. 1–11.
[39] J. Sexton, "Automatic CNN based Enhancement of 360 Video Experience with Multisensorial Effects," 2021. [Online]. Available: https://github.com/sextonj5/Automatic-CNN-based-Enhancement-of-360-Video-Experience-with-Multisensorial-Effects
[40] J. Sexton, "Modified Categories of Places 365 Labels to Support Olfaction," 2021. [Online]. Available: https://raw.githubusercontent.com/sextonj5/MulseRepos/master/categories_places365.txt
[41] D. Richards, "Compatibility of 48 and 24Hz Motion Images: A Problem and a Solution," in Proc. Society of Motion Picture and Television Engineers Technical Conference and Exhibition (SMPTE), Oct. 2008, pp. 400–408.
[42] O. M. Bouzid, G. Y. Tian, J. Neasham, and B. Sharif, "Investigation of Sampling Frequency Requirements for Acoustic Source Localisation Using Wireless Sensor Networks," Applied Acoustics, vol. 74, no. 2, pp. 269–274, Feb. 2013.
[43] "360° Great Hammerhead Shark Encounter | National Geographic ," 2016. [Online]. Available: https://www.youtube.com/watch?v=rG4jSz_2HDY
[44] "Malaekahana Sunrise," 2015. [Online]. Available: https://www.youtube.com/watch?v=-bIrUYM-GjU&t
[45] "Ocean 360° - 4K Nature Meditation for Daydream, Oculus, Gear VR ," 2017. [Online]. Available: https://www.youtube.com/watch?v=xLf002jIko8
[46] "Discovery VR Walk Among the Giants ," 2015. [Online]. Available: https://www.youtube.com/watch?v=DTlzIGFrL_4
[47] "Experience Amsterdam: A Guided City Tour (360 VR Video) ," 2016. [Online]. Available: https://www.youtube.com/watch?v=FzrkpXlRP1M
[48] "Lions 360° | National Geographic," 2017. [Online]. Available: https://www.youtube.com/watch?v=sPyAQQklc1s
[49] "Glow Worm Caves of New Zealand in 360° | National Geographic," 2016. [Online]. Available: https://www.youtube.com/watch?v=QjqGlLVIAtg
[50] "360° Antarctica - Unexpected Snow | National Geographic," 2016. [Online]. Available: https://www.youtube.com/watch?v=XPhmpfiWEEw
[51] "360° Bryce Canyon | National Geographic," 2017. [Online]. Available: https://www.youtube.com/watch?v=t3gur-osvzY
[52] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights," in Proc. International Conference on Learning Representations (ICLR), Apr. 2017.
[53] V. T. Tran, N. H. Le, P. T. Nguyen, and T. N. Doan, "Lightweight Network for Vietnamese Landmark Recognition based on Knowledge Distillation," in Proc. NAFOSTED Conference on Information and Computer Science (NICS), Nov. 2020, pp. 286–291.

JOHN PATRICK SEXTON received a First Class Honors B.Eng. in Electronic and Computing Engineering from Dublin City University (DCU), Ireland. He is currently working in a Software and Computer Engineering role in Photolithography for Nikon Precision Europe GmbH. He is highly interested in novel applications of machine learning, such as Virtual Reality and Mulsemedia.



ANDERSON AUGUSTO SIMISCUKA (S'17-M'20) received the B.Sc. degree in Information Systems in 2014 from Mackenzie Presbyterian University, São Paulo, Brazil, and the Ph.D. degree from the School of Electronic Engineering, Dublin City University (DCU), Ireland, in 2020. He has worked in several telecom and software development projects in companies such as Wittel (2010-2013), DCU/Ericsson (E-Stream Project, 2014), Arkadin (2014) and IBM (2015). He is currently a Postdoctoral Researcher with the Performance Engineering Laboratory and the Insight SFI Centre for Data Analytics, School of Electronic Engineering, DCU. He is involved in the EU Horizon 2020-funded project TRACTION. He is a member of the IEEE Young Professionals, IEEE Communications Society, and IEEE Broadcast Technology Society.

KEVIN MCGUINNESS is an Assistant Professor with the School of Electronic Engineering in Dublin City University and a Science Foundation Ireland Funded Investigator at the Insight SFI Centre for Data Analytics and in the ENABLE research program. He graduated from Dublin City University in 2005 with a BSc (Hons) in Computer Applications and Software Engineering. He subsequently joined the Centre for Digital Video Processing group in DCU, and was awarded a PhD from the School of Electronic Engineering in 2009. Kevin has since been employed as a postdoctoral researcher in CLARITY: Centre for Sensor Web Technologies in Dublin City University, and a research fellow at the Insight Centre for Data Analytics.

GABRIEL-MIRO MUNTEAN (M'04–SM'17) is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and co-Director of the DCU Performance Engineering Laboratory. He has published over 400 papers in top-level international journals and conferences, authored 4 books and 23 book chapters, and edited 8 additional books. He is an Associate Editor of the IEEE Transactions on Broadcasting, the Multimedia Communications Area Editor of the IEEE Communications Surveys and Tutorials, and a Reviewer for important international journals, conferences, and funding agencies. He was the Project Coordinator for the EU-funded project NEWTON http://www.newtonproject.eu and is the DCU Coordinator for the EU project TRACTION https://www.traction-project.eu.