xR4DRAMA

Extended Reality For DisasteR management And Media planning

H2020-952133

D3.2

Outdoors localization algorithms & tools v1

Dissemination level:	Public
Contractual date of delivery:	Month 12, 31.10.2021
Actual date of delivery:	Month 13, 09.11.2021
Work package:	WP3 - Analysis and fusion of multi-modal data
Task:	T3.2 Visual analysis
Type:	Demonstrator
Approval Status:	Final Version
Version:	1.0
Number of pages:	39
Filename:	d3.2_xR4Drama_outdoorslocv1_20210906_v1.0.pdf

Abstract

This deliverable describes initial version of exterior localization algorithms & tools.

co-funded by the European Union

History

Version	Date	Reason	Revised by
V0.1	06.09.2021	Table of contents	CERTH
V0.2	22.10.2021	1st version of the deliverable	CERTH
V0.3	04.11.2021	2nd version of the deliverable for internal review	CERTH
V1.0	09.11.2021	Final document to be submitted	CERTH

Author list

Organization	Name	Contact Information
CERTH	Spyridon Symeonidis	spyridons@iti.gr
CERTH	Haralabos Papadopoulos	chapapadopoulos@iti.gr
CERTH	Sotiris Diplaris	diplaris@iti.gr
CERTH	Anastasios Karakostas	akarakos@iti.gr

Executive Summary

This deliverable reports on the basic techniques for outdoors localization algorithms and tools. Specifically, it elaborates on the initial methods for: (i) Shot Detection (SD), (ii) Scene Recognition (SR), (iii) Emergency Classification (EmC), (iv) Photorealistic Style Transfer (PST) and (v) Building and Object Localization (BOL). The goal of the Building and Object Localisation (BOL) component is to analyse the compiled visual data (images, videos) and provide on the one hand a semantic meaningful annotation for the observed visual objects and scenes and on the other hand a computational boost to 3D reconstruction. Photorealistic style transfer aims to assist building and object localization component so that we get more accurate results.

The document elaborates on the WP3 modules, which are related to T3.2 and the appropriate approaches, components, and resources that were adopted to fulfil the respective functionalities that were described in the Description of Actions (DoA) and later on documented by the users throughout the compiled user requirements (D6.1, D6.2). The deliverable introduces the basic techniques for shot detection, scene recognition, emergency classification, photorealistic style transfer and building and object localization that were deployed during the first phase of the project's lifetime, for the implementation of the 1st prototype (M13). Furthermore, a description of the analysis requirements for visual components is provided and analysed thoroughly, while for each module an overview of the State-of-the-Art (SoA) and a comparison to other approaches is documented. The evaluation approaches and results are finally explained and demonstrated at the end of the document.

More specifically, the modules that are described in further detail are the ones that were deployed for fulfilling the basic functionalities of BOL component:

i. The **Shot Detection (SD)**, which analyses visual data in order to segment the acquired video in shots and extract the most meaningful frames (i.e. keyframes) for further analysis.
ii. The **Scene Recognition** (**SR**), which provides a high-level annotation about the type of area or buildings that are depicted in the visual scenes.
iii. **Emergency Classification (EmC),** which detects flood or fire that may exist in the analysed images or videos.
iv. **Photorealistic Style Transfer (PST),** which generates new images with the same content and the style of a selected image that can be transferred to make them look like they are in different lighting, time of day or weather. It aims to provide enhanced input images to object detection and localisation algorithm, so as to get more accurate results.
v. **Building and Object Localisation (BOL),** which is responsible to detect, recognize and localize buildings and the desired buildings, objects or elements that might exist in the acquired xR4DRAMA image and video samples.

It is worth to note, that the performance of the above modules has been extensively evaluated in terms of their accuracy and the first experimental results are encouraging to continue to work on this direction.

Abbreviations and Acronyms

BOL	Building and Object Localisation
CNN	Convolutional Neural Networks
ConvNets	Convolutional Neural Networks
DCNNs	Deep Convolutional Neural Networks
EmC	Emergency Classification
EmL	Emergency Localization
IoU	Intersection over Union
mIoU	mean Intersection over Union
PNST	Photorealistic Neural Style Transfer
PST	Photorealistic Style Transfer
PUC	Pilot Use Case
ReLU	Rectified Linear Unit
SD	Shot Detection
SoA	State of the art
SR	Scene Recognition
SVM	Support Vector Machine
WCT	Whitening and Colouring Transform

Table of Contents

1 INTRODUCTION

In xR4DRAMA, the scope of T3.2 - "Visual analysis" is to localise the exteriors of buildings, the surrounding environment, as well as objects and other valuable assets needed for media production and situation monitoring. The extracted subsequences are used to assist the construction of the 3D-model of an unknown area, including the detected objects of interest (T4.4). Moreover, the meaningful semantic information emerges from the image or video analysis that enriches xR4DRAMA's Knowledge Base (KB) (T3.5). In general, the modules of T3.2 are essential and useful tools that help on the acceleration and semantic augmentation of the xR4DRAMA 3D models.

During the first half of xR4DRAMA project lifetime (M1-M13), T3.2 contributed on the first milestone (MS1) by defining the technical requirements of its modules (D5.1) and aligning them towards the user requirements (D5.2). T3.2 also contributed to the second milestone (MS2) of the xR4DRAMA project by deploying and integrating the initial versions of Shot Detection (SD), Scene Recognition (SR), Emergency Classification (EmC), Photorealistic Style Transfer (PST) and Building and Object localisation (BOL) to the 1st prototype of the system. The iteration of T3.2's implementation will continue until the completion of the final prototype, by contributing to the final prototype (MS4). The described timeline is depicted in *Figure 1*.

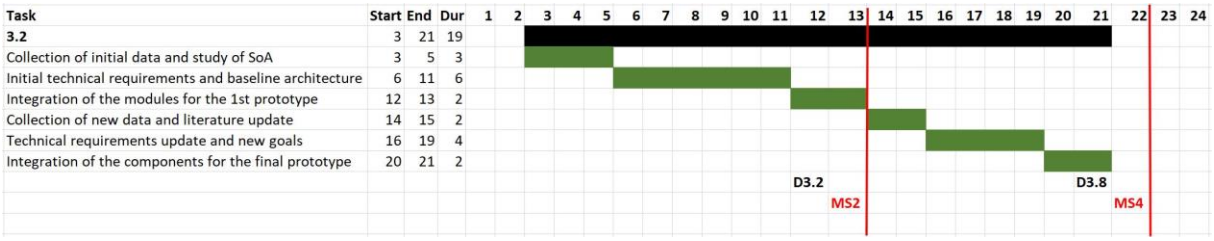

Figure 1: Timeline of T3.2

1.1 Objectives

The objectives of T3.2 for the 1st period of the project (M1-M13) are aligned with the main goals that were described in the DoA and summarized as follows:

- Study the literature that exists on shot detection, scene recognition, emergency classification, photorealistic style transfer and building and object localization (*Accomplished*).
- Design and deploy the appropriate computer vision and deep learning algorithms that will recognise the type of area or buildings depicted in images or videos and also determine whether further processing of them will be useful (e.g., if an indoor scene is detected there will be no further processing of the corresponding images or video frames). *(Accomplished by implementing the SD and SR modules)*
- Develop a proper deep learning algorithm for the recognition of emergency events (e.g., flood or fire) in images or videos. *(Accomplished by implementing the EmC module)*
- Develop proper image semantic segmentation algorithms for the localisation of building and their surroundings, objects and other elements. *(Accomplished by the implementation of the BOL module)*

T3.2 also fulfilled several other goals, in order to satisfy the xR4DRAMA's use cases and user requirements (D6.1, D6.2):

- Not only distinguish whether buildings or other objects of interest exist inside image and video frames, but also identify the scene where they exist. An initial estimation provides knowledge on whether the depicted scene is from an interior or exterior environment, while a more detailed estimation is given afterwards, predicting the class of the place. *(SR module)*
- Distinguish between foreground and background pixels within a video frame or image, estimated that it contains a building, in order to facilitate the 3D reconstruction process (T4.4). *(BOL module)*
- Detect and provide information about 19 semantic classes. (*BOL module*)
- Determine the existence of flood or fire in analysed images or videos. (*EmC module*)
- Diminish sensitive information (e.g., people, car licence plates' number) or moving objects (e.g., vehicle) from images or video frames in order to assist the 3D reconstruction process of an unknown area or building through archival videos. *(BOL module)*

1.2 Results towards the foreseen objectives of the xR4DRAMA project

Until now, xR4DRAMA has fulfilled the foreseen objectives of the project by completing the development of the basic functionalities of shot detection, scene recognition, emergency classification and building and object localisation with the following activities:

a) Gathered annotated visual data from benchmark datasets (e.g., Places-2[1], SUN[2], CityScapes[3]) and xR4DRAMA consortium partners and used them to train the Scene Recognition (SR), Emergency Classification (EmC) and Building and Object Localization (BOL) models.
b) Deployed the initial version of Shot Detection (SD) for videos by using SoA computer vision and deep learning algorithms.
c) Deployed the initial version of Scene Recognition (SR) in images and video frames using SoA computer vision and deep learning scene recognition algorithms in the compiled datasets.
d) Deployed the initial version of Emergency Classification (EmC) using SoA image classification algorithms.
e) Deployed the initial version of Photorealistic Style Transfer (PST).
f) Deployed the initial version of Building and Object Localisation (BOL) in images and video frames by using SoA computer vision and deep learning image semantic segmentation algorithms.

[1] http://places2.csail.mit.edu/download.html

[2] https://groups.csail.mit.edu/vision/SUN/hierarchy.html

[3] https://www.cityscapes-dataset.com/

1.3 Future Plans

- To re-study the literature that exists on shot detection, scene recognition, emergency classification, photorealistic style transfer and building and object localization in order to update any advances and improvements that have been introduced.
- To gather additional visual annotated material and datasets to enhance the classification and segmentation models that have been developed.
- To accelerate the computational efficiency of the visual analysis modules by redesigning and compressing the corresponding deep learning architectures.
- To extend building and object localisation for videos by tracking and maintaining the coherency of detected objects and buildings throughout time.
- To extensively test the combination of PST and BOL modules so that we successfully localise buildings and objects even in images with difficult weather or lighting conditions (e.g., rain, fog or night).

1.4 Outline

The outline of this deliverable is as follows. Sections 2 and 3 respectively contain a brief presentation of the relevant user requirements for the analysis of the visual content and a description of the relevant SoA methodologies in the scientific fields of computer vision, deep learning and segmentation. The methodology analysis of the visual analysis modules, Shot Detection (SD), Scene Recognition (SR) and Emergency Classification (EmC), Photorealistic Style Transfer (PST) and Building and Object Localization (BOL) are then described in Sections 4 and 5. Parameter selection and evaluation metrics for all deployed algorithms is provided in Section 6, while Section 7 concludes the deliverable and defines the future work the visual analysis modules until M21.

2 OUTDOORS LOCALIZATION REQUIREMENTS

The xR4DRAMA user requirements have been initially reported in D6.1 “Pilot use cases and initial user requirements” and finally defined in D6.2 “Final user requirements”. *Table 1* presents the nine user requirements reported in D6.2 that are related to the visual analysis components. All, except PUC1-08, are associated with Scene Recognition (SR), Emergency Classification (EmC) and Building and Object Localization (BOL). Further details are provided in Sections 2.1. and 2.2. PUC1-08 requires the development of an extra module named Visual River Sensing (VRS) for the detection of river overtopping, which will take place during the next phase of the project. It will be clear from this section, that T3.2 is aligned with the user requirements that were defined in D6.2, as the basic versions of its modules (SD, SR, EmC, PST and BOL) already satisfy the defined needs.

Table 1: Relevant user requirements reported in D6.2 for the visual analysis components

User Requirement (UR)	Category	Name	Description	Priority (1=high, 4=low)
G-01	Accessibility	Transportation	quality and type of road (highway, street, path), distance to railway station and airport, public transport	3
G-02	Geography, Surroundings	Buildings, Monuments	the shape, look and size of buildings, the purpose of buildings	1
G-03	Geography, Surroundings	Landmarks	indication of high voltage lines, windmills and other landmarks	1
G04	Geography, Surroundings	Roads, Railroads	indication of roads, highways, railroads	1
PUC1-07	Flood risk management	Flooded elements	Information on flooded elements (e.g., cars and people inside the river)	1
PUC1- 08	Flood risk management	River embankment's overtopping and/or breaking	Information related river embankments	1

			overtopping or breaking	
PUC1- 09	Flood risk management	Elements at risk	Information on the presence of elements at risk and the degree of emergency	1
PUC2-01	Environmental factors	Noise pollution	identification of possible sources like busy roads or highways, crowds of people, factories, airports, railway stations, railway tracks	1
PUC2- 02	Environmental factors	Light Pollution	identification of possible sources like streetlights, ads etc.	2

2.1 Scene Recognition and Emergency Classification Requirements

As can be seen in *Table 1* most of the user requirements from D6.2 have been identified that can be directly or indirectly associated with the Scene Recognition (SR) and Emergency Classification (EmC) modules of xR4DRAMA, namely G-01, G-02, G-03, G04, PUC2-01, PUC2-02, PUC1-07 and PUC1-09.

As far as G-01, G-02, G-03, G04, PUC2-01 and PUC2-02 are concerned, users required from the xR4DRAMA platform to extract meaningful tags and semantics data from the acquired video and image samples. Scene Recognition (SR) is a candidate module that can satisfy these criteria by providing meaningful information about the type of area (e.g., downtown, village, airfield), road (e.g., driveway, highway, alley) or building (e.g., church, castle, palace). Moreover, area's landmarks can be recognized, such as windmills. This data can be saved in xR4DRAMA's Knowledge Base (KB).

In PUC1-07 and PUC1-09 users defined that they would like to get information about flooded elements and elements at risk. For this to be possible, the Emergency Classification module will identify possible flood in the analysed images or videos.

2.2 Building and Object Localization Requirements

The same user requirements from D6.2, namely G-01, G-02, G-03, G04, PUC2-01, PUC2- 02, PUC1-07 and PUC1-09, can also be directly or indirectly associated with the Building and Object Localization (BOL) module of xR4DRAMA.

For G-01, G-02, G-03, G04, PUC2-01 and PUC2-02 that require from the xR4DRAMA platform to extract meaningful tags and semantics data from the acquired video and image samples. Building and Object Localisation module can satisfy these criteria by providing meaningful information about the existence of buildings, constructions (e.g., tunnel, bridge) and surrounding elements (e.g., streetlights, traffic signs, cars).

Concerning PUC1-07 and PUC1-09 where users defined that they would like to get information about flooded elements and elements at risk, the Building and Object Localization module can localise vehicles and people that are in danger.

The above user requirements apply also to the Photorealistic Style Transfer (PST) module, as its aim is to ameliorate the results of the Building and Object Localization (BOL) module and is an important step before the localisation algorithm.

3 RELEVANT WORK

During the first months of the first period (M3-M5), a thorough study of the relevant computer vision and deep learning domain has taken place. The study was mainly concentrated on the algorithms that were foreseen to been implemented during M3-M13, meaning Shot Detection (SD), Scene Recognition (SR), Emergency Classification (EmC), Photorealistic Style Transfer (PST) and Building and Object Localization (BOL). This involved the study of Deep Learning and Computer Vision algorithms that focus on shot detection, scene recognition, emergency classification, photorealistic style transfer and semantic segmentation.

3.1 Shot detection

Automatic shot boundary detection is the complete segmentation of a video into continuously imaged temporal video segments by identifying shot transitions (e.g., fade in/out, cuts, etc.). The challenge for the shot boundary detection algorithms is to successfully detect these shot transitions, as fast abrupt changes of the visual contents caused by camera/environment/entity can still confuse even SoA models and may lead to many false hits. Besides false hits, trained models can sometimes miss a difficult (e.g., long) transition, even if such type was present in the utilized train set. In addition, novel "wild" transition types supported in common video editing tools and often used in television content may find shot transition detection models "unprepared" for a given type.

Currently, end-to-end deep learning approaches have become a mainstream research direction for video analysis tasks. For transition detection, temporal context is essential, and so a detection approach requires either an aggregation of extracted features from individual frames (Karpathy, et al., 2014) or utilization of 3D convolutions that jointly process spatial and temporal information. The latter approach, popularized by (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015) for video classification, was also considered for shot boundary detection networks proposed by (Hassanien, Elgharib, Selim, Hefeeda, & Matusik, 2017) and (Gygli, 2017). (Hassanien, Elgharib, Selim, Hefeeda, & Matusik, 2017) predict a likelihood of sharp or gradual transition in a 16-frame sequence by the C3D network (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015). The predictions are, however, not used directly, and an SVM classifier is trained to give a labelling estimate. Further, some false positives are suppressed by colour histogram differencing. (Gygli, 2017) on the other hand, utilises only predictions from a 3D convolutional network without any post-processing. However, the network is much smaller and outperformed by (Hassanien, Elgharib, Selim, Hefeeda, & Matusik, 2017). TransNet (Lokoč, Kovalčik, Souček, Moravec, & Čech, 2019), a 3D convolutional network combines the end-to-end no-post-processing approach of (Gygli, 2017) with performance comparable to (Hassanien, Elgharib, Selim, Hefeeda, & Matusik, 2017) on RAI dataset (Baraldi, Grana, & Cucchiara, 2015). A new improved version of TransNet architecture, named TransNet V2, is proposed in (Souček & Lokoč, 2020), which provides promising detection accuracy and enables efficient processing of larger datasets.

3.2 Scene recognition and Emergency Classification

In this subsection, we present a study of the scene recognition and emergency classification related literature. Scene recognition aims at the categorisation of the scenes depicted in

images or videos, while emergency classification recognises emergency situations, like flood or fire, in images or videos. Both are subdomains of image classification.

VGG16, VGG19 (Krizhevsky, 2012) and ResNet DCNN (He K. Z., 2016) are very famous architectures for image classification. (Gong, 2014) applies convolutional neural networks (ConvNets) within local multi-scale patches and integrates the patch-based ConvNets with global ConvNets, in order to capture both detailed information and holistic characteristics in scenes. Moreover, in (Gangopadhyay, 2016), a statistical aggregation solution is proposed, based on ConvNets for scene classification. Both the ConvNets on large datasets (to acquire spatial information) and the resulting ConvNets features were further analysed by statistical methods in the temporal domain to maintain spatio-temporal coherency throughout their representation. The acclaimed C3D feature was proposed in (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015), and describes how to transform 2D ConvNets to 3D ConvNets in order to exploit deep convolutional information in both spatial and temporal dimensions. However, C3D can only handle small video clips with a few frames and discards the long-term information in videos, due to the large computational cost. (Huang, 2019) focus, firstly, on the short-term motion and spatial properties, and secondly, on the long-term motion information. In this way, the method combines long-term information with short-term deep information, in order to obtain a complementary representation and better understanding toward scene recognition.

For scene recognition, (Zhou B. L., 2017) introduced a scene-centric dataset called Places with more than 7 million images of scenes and focus on representing images through a holistic approach, trying to recognize the scene/place instead of separate objects. SUN397 (Xiao, Hays, Ehinger, Oliva, & Torralba, 2015) is another scene recognition related dataset that contains 908 scene categories with a varying number of images per category.

As for the emergency classification problem, the VRBagged-Net framework (Muhammad, Tahir, & Rafi, 2021) is proposed and implemented for flood classification. The framework utilizes the deep learning models Visual Geometry Group (VGG) and Residual Network (ResNet), along with the technique of Bootstrap aggregating (Bagging).

In Sections 4.2 and 4.3, we describe in detail the methodologies of the implemented xR4DRAMA approaches for scene recognition and emergency classification that are based on the VGG-16 architecture.

3.3 Photorealistic style transfer

Photorealistic neural style transfer (PNST) is another category of neural style transfer that aims to generate output images that still look like a real shot after the style transfer, and in parallel, it involves all textures and patterns that characterise the given style. One of the most influential works in photorealistic neural style transfer is the work of (Luan, Paris, Shechtman, & Bala, 2017), in which they propose a regularisation in the objective function during the optimisation process. The main constraint is that the reconstructed image is represented by locally affine colour transformations of the input to prevent distortions, and they use a guidance to the style transfer process based on semantic segmentation of the inputs for content preservation.

Another PNST work, that also improves the photorealism of the results, has been introduced by (Gatys, 2017) with a set of spatial, colour and scale controls in the image stylisation process. These control methods are also applicable to fast approximations of Neural Style Transfer. (He M. , 2017) propose a colour transfer approach between images that involves neural feature representation for semantic matching. To avoid inconsistencies and non-local constraints, their local colour transfer in the image domain applies a linear transform at every pixel.

Moreover, a whitening and colouring transform (WCT) has been proposed in (Li Y. e., 2017), which matches the feature covariance of the content image to a given style image. PhotoWCT (Li Y. e., 2018) leverages unpooling in the model structure and applies a second post-processing step for optimal final results. This post-processing step provides more effective results but at a higher computational cost. WCT^2 (Yoo, 2019) addresses this problem by replacing the unpooling with wavelet pooling and unpooling that improve the quality of stylised images skipping the post-processing step. A linear propagation module is introduced in (Li et al., 2019) that enables a feed-forward network in photorealistic neural style transfer. The proposed framework has two feed-forward networks, a symmetric encoder-decoder image reconstruction part, and a transformation learning part, using a light-weighted CNN block. (Penhouët, 2019) propose a modification of the stylised image by improving the aesthetic quality features of the image with the use of Neural Image Assessment.

(Kurzman, 2019) introduce a Class-Based Styling method that map different styles for different object classes efficiently. Firstly, a semantic segmentation method is leveraged to obtain the mask of each object class, and secondly, a style transfer method is used to stylise the image or video frame. For semantic segmentation, they use DABNet and for the style transfer step, they consider the method proposed by Johnson (Johnson, Alahi, & Fei-Fei, 2016).

Recently, (Xia, 2020) has introduced a feed-forward neural network that learns local edge-aware affine transforms with the photorealism constraint. The network is based on HDRnet, which was initially introduced in image enhancement and tone manipulation. The learned transforms are intentionally constrained to preserve content without artifacts such as noise or false edges. Finally, (An, 2020) have presented a framework that consists of a construction step (C-step) with an auto-encoder named PhotoNet, to build a photorealistic stylisation network, and then a pruning step (P-step) to become more efficient.

3.4 Building and object localization

Deep learning techniques have enormous success solving both image classification and segmentation problems. Image semantic segmentation has the goal to assign semantic labels to every pixel in the analysed image. Fully Convolutional Networks for Semantic Segmentation, presented by (Long, 2015), popularized the use of end-to-end convolutional networks and introduced skip connections from higher resolution feature maps. (Lin, 2017) propose to use an encoder part of ResNet-101 (He K. Z., 2016) blocks and a decoder part of RefineNet (Lin, 2017) blocks, which concatenate high-resolution features from encoder and low-resolution features from previous RefineNet blocks. Another encoder-decoder architecture was proposed by (Peng, 2017) which includes very large kernels convolutions, but these large kernels convolutions are computationally expensive and they are adopted because networks tend to gather information from a smaller region. DeepLabV2 network (Chen L.-C. a., 2017) is an architecture for semantic segmentation that builds on DeepLab (Chen L.-C. ,

2014) with an atrous spatial pyramid pooling scheme. New versions of it have been proposed, DeepLabV3 (Chen L.-C. e., 2017), which improves upon DeepLabv2 with several modifications, and DeepLabV3+ (Chen et al, 2018), which, in turn, extends the previous one.

4 SCENE RECOGNITION AND EMERGENCY CLASSIFICATION V1

In this section, we describe the methodologies followed for the development of the initial versions of Shot Detection (SD), Scene Recognition (SR), Emergency Classification (EmC), Photorealistic Style Transfer (PST) and Building and Object Localisation (BOL).

4.1 Shot detection

In many cases, videos and documentaries, directed by professionals, contain shots of multiple scenes. Some of these scenes may not be directly relevant to the information we would like to extract or useful for 3D reconstruction purposes. It is, therefore, important to pre-process these videos so that we split them in the different scenes that they contain to examine if further processing of them will be useful for the purposes of the project. In a preliminary step, the various shots are first delineated and can then be further processed.

For this purpose, we have deployed TransNet V2 (Souček & Lokoč, 2020), a deep network that reaches SoA performance on respected benchmarks. TransNet V2[4] is based on the original TransNet concepts (Lokoč, Kovalčik, Souček, Moravec, & Čech, 2019), where a resized input sequence of frames is initially processed with Dilated DCNN cells. Specifically, the previously released TransNet version comprises six DDCNN cells where every cell consists of four *3 × 3 × 3* convolution operations, each with *F* filters and different dilation rates *1, 2, 4, 8* for the temporal dimension. Hence, a larger receptive field of *97* frames is reached by the final sixth TransNet's DDCNN cell while using still an acceptable number of learnable parameters. In the new version, DDCNN cells also incorporate batch normalization that stabilizes gradients and adds noise during training. Every second cell contains a skip connection followed by spatial average pooling that reduces spatial dimension by two with additional improvements. In *Figure 2*, the architecture of TransNet V2 is presented.

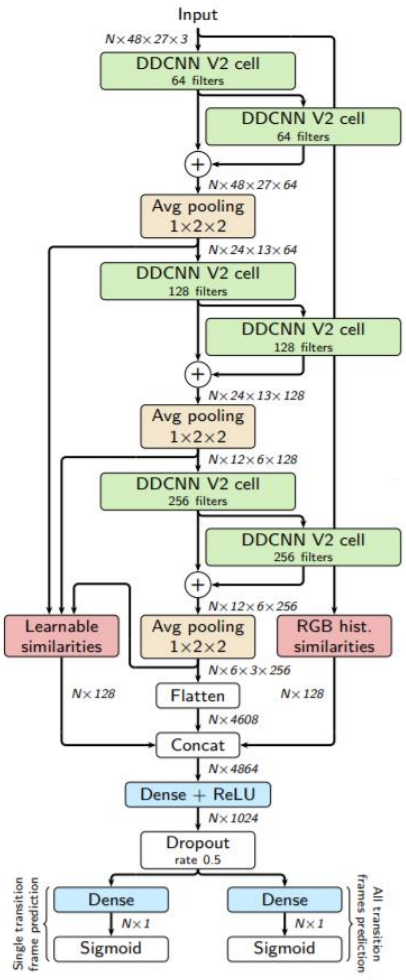

Figure 2:
The TransNet V2 architecture
(Souček & Lokoč, 2020)

4.2 Scene recognition

As far as scene recognition is concerned, we use Deep Convolutional Neural Networks (DCNNs) with two components: one on the hidden layers for the feature extraction part, and

[4] https://github.com/soCzech/TransNetV2

one for the classification part. In the feature extraction component, the network combines a sequence of convolution and pooling operations, where the features are progressively detected. In the classification part, the fully connected layers serve as a classifier on top of these extracted features, assigning a probability for each class that the algorithm predicts.

Convolution is one of the main operations in a DCNN architecture, being the mathematical combination of two tensors to produce a third one. The convolution is performed on the input data with the use of a filter (known also as kernel) to then produce a feature map. We execute a convolution by sliding the filter over the input, which can be either a 2D or 3D array of elements. At every location, a matrix element-wise multiplication is performed and the result is summed onto the feature map. The output of the convolution is passed through an activation function. Stride is the step of the convolution filter displacement for each step and it is usually equal to one, meaning that the filter slides pixel by pixel.

In general, the size of the feature map is always smaller than the input; hence, it is common to prevent the feature map from shrinking using padding. After one or a stack of convolution layers, it is common to add one pooling layer to continuously reduce the dimensionality, thus reducing the number of parameters, to decrease the training time. The most frequent type of pooling is max pooling, which takes the maximum value in each considered window.

The convolution and pooling layers are then followed by a few fully connected layers (FC), which can only accept one-dimensional data. To convert our 3D feature array to one-dimensional vector we "flatten" the array by concatenating the rows of each dimension. This vector is further passed to a logistic regression classifier to produce the final vector of class score predictions. The input size of our training set is a set of m images with dimensions $n_h * n_w * \mathrm{n_c}$, where n_h and n_w are the height and the width of an image with n_c channels. VGG16 is a 16-layer neural network, not including the max-pool layers and the SoftMax activation in the last layer. In particular, the image is passed through a stack of convolutional layers, which are used with filters of a small receptive field $f * f$. Spatial pooling is carried out by five max-pooling layers, which follow some of the convolutional layers (not all), as described in the original paper (Simonyan K. a., 2014). Max-pooling is performed over a $w_p * w_p$ pixel window, with stride s.

The width of convolutional layers starts from 64 in the first layers and then increases by a factor of 2 after each max-pooling layer, until it reaches 512 as depicted in Figure 3. The stack of convolutional layers is followed by three Fully-Connected (FC) layers. The final layer is the SoftMax layer. All hidden layers are equipped with the Rectified Linear unit (ReLU (Krizhevsky, 2012)), which is defined in *Equation 1*.

$$f(x) = -\max(0, x) \tag{1}$$

ReLU is an element-wise operation, applied per pixel, and replaces all negative pixel values in the feature map with zeros. The main property of ReLU is the introduction of non-linearity of the Convolutional Network and therefore the ability to identify and extract realistic non-linearity.

In the context of xR4DRAMA's Scene Recognition (SR) module, the VGG16 framework was pre-trained on Places dataset[5] on first 14 layers, which has the initial 365 Places categories. The remaining layers were trained on a subset of 99 selected classes of Places dataset presented in the *Table 3,* in order to adjust the SR model to the xR4DRAMA needs and classify only the relevant scenes. Moreover, the Scene Recognition (SR) model classifies scenes in two general environmental categories, i.e. indoor or outdoor, and use them so as to differentiate between frames that could be furtherly processed and used for 3D reconstruction of an area (T4.4)(outdoor scenes) or not (indoor scenes).

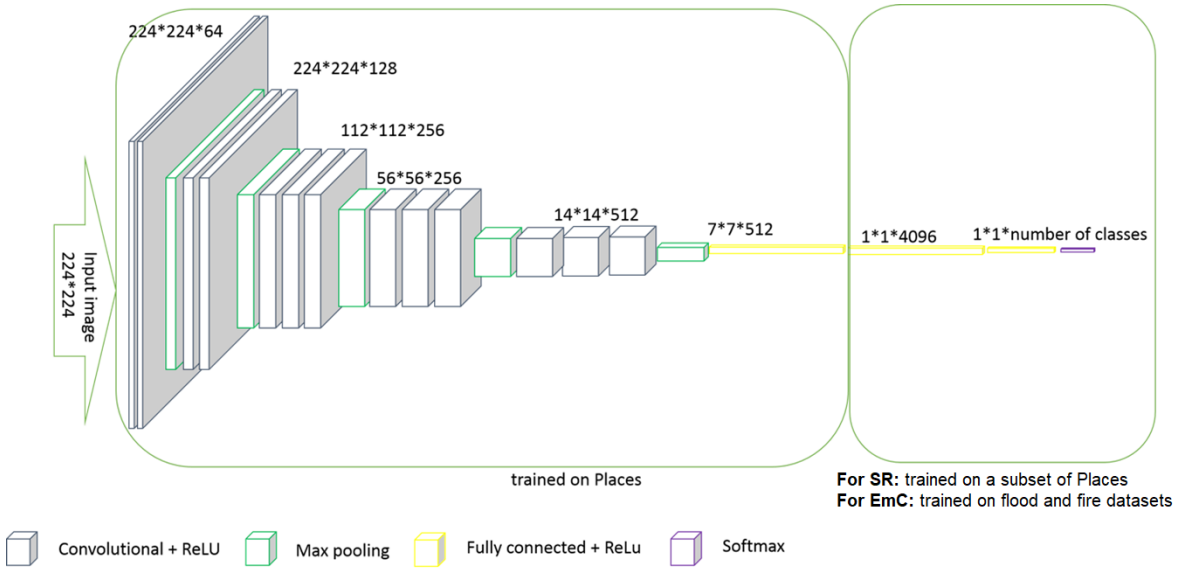

Figure 3: The VGG-16 architecture used for SR and EmC models.

4.3 Emergency classification

The Emergency Classification (EmC) module is also based on SoA image classification techniques and is used so as to determine which images contain an emergency event. Inspired from the recent success that deep learning showed in image understanding (Simonyan & Zisserman, 2014) and scene recognition (Zhou, Lapedriza, Xiao, Torralba, & Oliva, 2014), fine-tuning of the pre-trained parameters of the VGG-16 on Places365 dataset was performed so as to leverage useful distinctions between various visual clues that relate to generic scenery images. We adapted its architecture so as to fit it the EmC purposes. Similarly to the SR framework, the final Fully Connected (FC) layer was removed and replaced with a new FC layer with a width of 3 nodes freezing the weights up to the previous layer and also a softmax classifier was deployed so as to enable multi-class recognition, as shown in *Figure 3*. More specifically the EmC results into three-class image recognition: "Flood", "Fire" and "Other", where "Other" may represent any theme except for fire and flood events.

The EmC results are integrated in the framework to indicate the existence of flood or fire events in a holistic manner and the component's purpose is to give an early indication and a first segment of solid information about the existence of an emergency in the image or video.

[5] http://places2.csail.mit.edu/

5 BUILDING AND OBJECT LOCALIZATION V1

5.1 Photorealistic style transfer

We use photorealistic style transfer to change an input image from night to day, from dark to light or from cloudy to sunny, while in parallel, the fine structures of the objects and buildings should remain intact. The purpose of this style transfer module is to increase the accuracy of object and building localisation in various luminance conditions. To achieve this we use a wavelet corrected transfer, based on the whitening and coloring transformation WTC^2 (Yoo, 2019) that allows features to preserve their structural information and statistical properties of the feature space during stylisation.

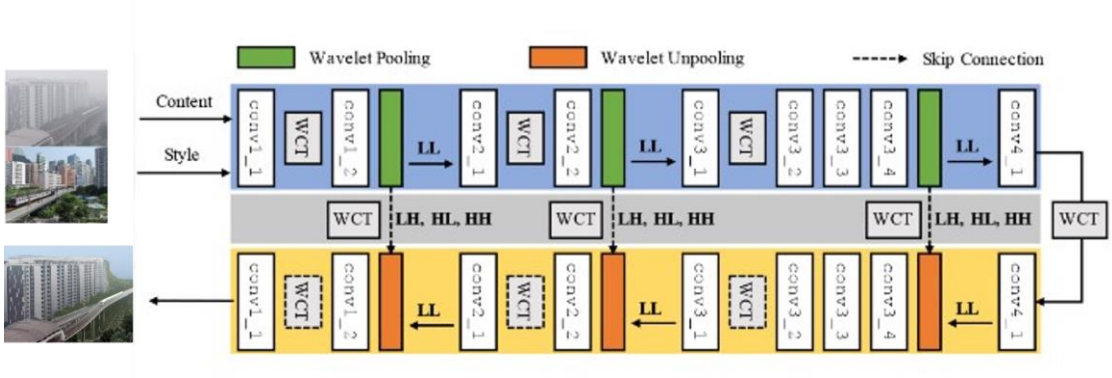

Figure 4: Overview of the progressive stylisation using the *WTC*^{2} model.

The model architecture of WTC^2 style transfer has a VGG-19 encoder with wavelet pooling and unpooling, and it is pre-trained on ImageNet, as it is illustrated in *Figure 4*. The decoder has a mirror structure of the encoder, and the Haar wavelet unpooling aggregates the components. Haar wavelet pooling has four channels; three of them refer to high frequency elements (vertical, horizontal, and diagonal edge-like information) and one on the low frequency domain, which captures smooth surface and texture information. The high frequency components (high-pass filters) are skipped to the decoder directly. Thus, only the low frequency component is passed to the next encoding layer as depicted in *Figure 5*. WTC^2 reconstructs correctly the input signal by mirroring its operation without post-processing steps, involving a minimal amount of information loss.

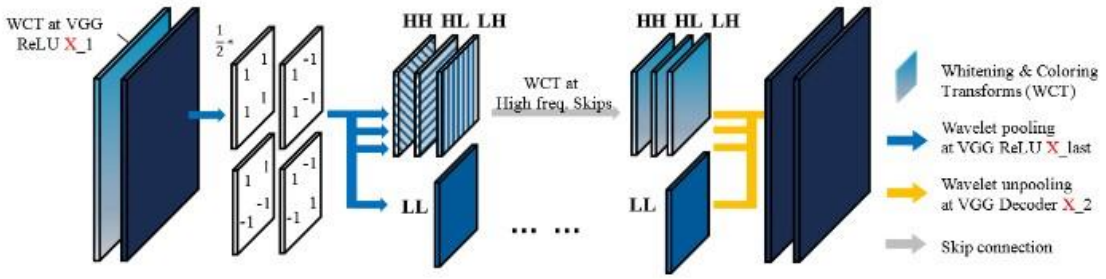

Figure 5: An encoder-decoder module using Haar wavelet pooling and unpooling.

Moreover, the wavelet features make the stylisation results from wavelet pooling more appealing. The usage of a single decoder during training and the inference time makes the

model be more efficient. Furthermore, due to the use of wavelet operations and progressive stylisation, the model amplifies errors when the multi-level strategy is applied.

5.2 Building and object localization

The Building and Object Localization (BOL) model of xR4DRAMA is based on DeepLabV3+ (Chen, 2018) that allows for segmenting images and visual content in general. The model is trained on the CityScapes dataset[6] adapting to the needs of xR4DRAMA. DeepLabv3+ extends DeepLabv3 by employing an encoder-decoder structure. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries.

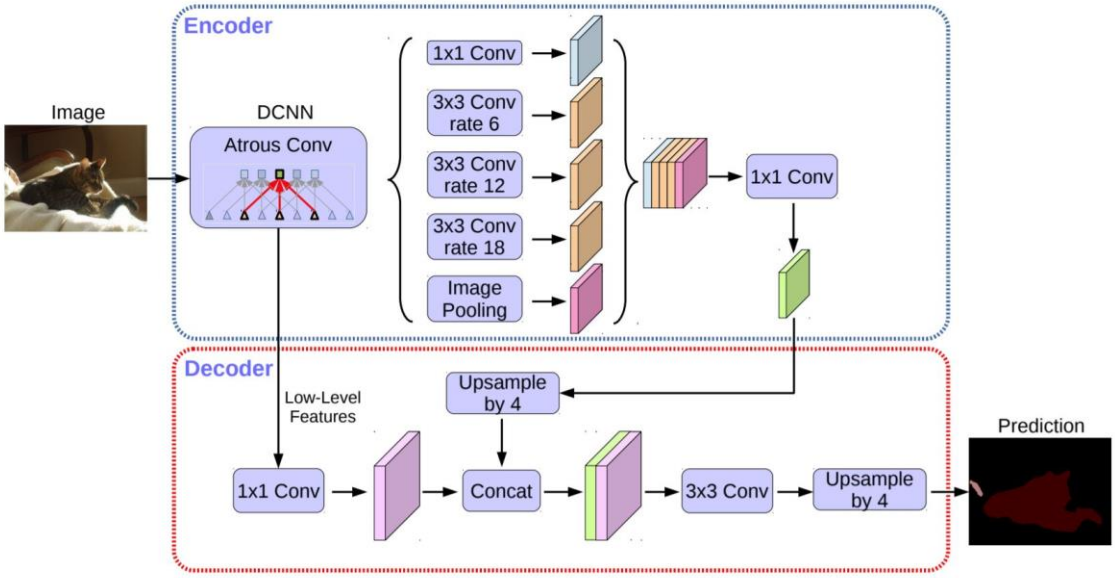

Figure 6: The DeepLabV3+ architecture (Chen, 2018).

A brief description of the basic network's elements follows:

- **Atrous convolution** generalizes the standard convolution operation. It is a powerful tool that allows the explicit control of the resolution of features computed by deep convolutional neural networks and the adjustment of filter's field-of-view in order to capture multi-scale information. In the case of two-dimensional signals, for each location i on the output feature map y and a convolution filter w, atrous convolution is applied over the input feature map x as follows:

$$y[i] = \sum_{k} x[i + r \cdot k]w[k] \qquad (2)$$

where the atrous rate r determines the stride with which we sample the input signal. Note that standard convolution is a special case in which rate *r = 1*. The filter's field-of-view is adaptively modified by changing the rate value.

[6] https://www.cityscapes-dataset.com/

- **Depthwise separable convolution**, factorizing a standard convolution into a depthwise convolution followed by a pointwise convolution (i.e., *1 × 1* convolution), drastically reduces computation complexity. Specifically, the depthwise convolution performs a spatial convolution independently for each input channel, while the pointwise convolution is employed to combine the output from the depthwise convolution.

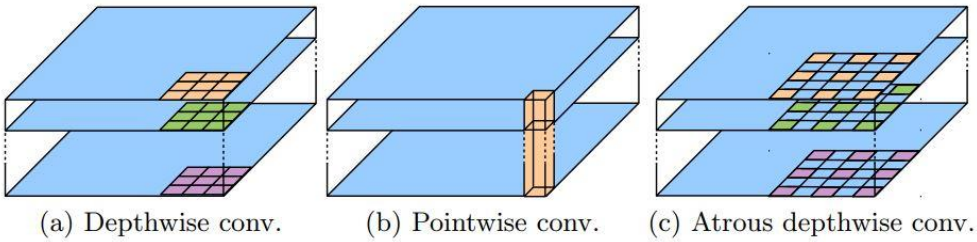

Figure 7: *3×3* Depthwise separable convolution decomposes a standard convolution into (a) a depthwise convolution and (b) a pointwise convolution (Chen, 2018).

- DeepLabv3 is the encoder network. It employs atrous convolution to extract the features computed by deep convolutional neural networks at an arbitrary resolution. For the task of semantic segmentation, one can adopt output stride = 16 (or 8) for denser feature extraction by removing the striding in the last one (or two) block(s) and applying the atrous convolution correspondingly. Additionally, DeepLabv3 augments the Atrous Spatial Pyramid Pooling module, which probes convolutional features at multiple scales by applying atrous convolution with different rates, with the image-level features (Liu, 2015).
- The encoder features are first bilinearly upsampled by a factor of *4* and then concatenated with the corresponding low-level features. There is *1x1* convolution on the low-level features before concatenation to reduce the number of channels, since the corresponding low-level features usually contain a large number of channels (e.g., *256* or *512*), which may outweigh the importance of the rich encoder features. After the concatenation, a few *3x3* convolutions are applied to refine the features followed by another simple bilinear upsampling by a factor of *4*.
- DeepLabV3+ uses ResNet-101 (He K. Z., 2016) or Modified Aligned Xception as backbone networks. Concerning Modified Aligned Xception network, the authors of DeepLabV3+ made a few changes on top of MSRA's work, namely (1) deeper Xception same as in (Haozhi, 2017) except that they do not modified the entry flow network structure for fast computation and memory efficiency, (2) all max pooling operations are replaced by depthwise separable convolution with striding, which enables the application of atrous separable convolution to extract feature maps at an arbitrary resolution, and (3) extra batch normalization (Sergey & Szegedy, 2015) and ReLU activation are added after each *3 × 3* depthwise convolution, similar to MobileNet design (Andrew, 2017).

In our approach, for our BOL model, we deployed the DeepLabV3+ architecture[7] with ResNet-101 as a backbone network and trained it on the CityScapes dataset[8].

5.3 Detection of people or vehicles in danger

In order to be able to detect people or vehicles in danger, we deployed a second image segmentation model in the BOL module, which can localise flood or fire regions inside images and videos. This Emergency Localization (EmL) model is based on a deep CNN technique for semantic image segmentation, fine-tuned by using water and fire image regions from publicly available datasets, so as to characterize the pixels of images as “water”, “flame”, or “background”. The resulting “water” and “flame” pixels would then form semantically localized areas inside an image where the corresponding emergency event had a direct impact. Once the “water” or “flame” regions are localised, we check if the BOL model localised any people or vehicles in this area too by comparing the masks that come from BOL and EmL models.

The Emergency Localization (EmL) model runs only if Emergency Classification (EmC) recognizes flood or fire event in the analysed images or videos.

[7] https://github.com/rishizek/tensorflow-deeplab-v3-plus

[8] https://www.cityscapes-dataset.com/

5.4 EVALUATION

In this section, we describe the datasets and settings that have been used for the development of the visual analysis components and we present corresponding analysis results.

5.5 Shot Detection

5.5.1 Dataset description

The TransNet V2 model[9] that we deployed for shot boundary detection is trained on the three following benchmark datasets:

1. The **RAI**[10] dataset, which mainly includes documentaries and talk shows. Shots and scenes have been manually annotated by a set of human experts to define the ground truth. For the shot detection task, the dataset contains 987 shot boundaries, 724of them being hard cuts and 263 gradual transitions.
2. The **BBC Planet Earth**[6] dataset that contains ground truth shots and scene annotation for each of the 11 episodes of the BBC Planet Earth educational TV Series. Each shot and scene has been manually annotated and verified by a set of human experts.
3. The **ClipShots**[11] , which is a large-scale dataset for shot boundary detection collected from Youtube and Weibo. It covers more than 20 categories, including sports, TV shows, animals, etc. Each video has a length of 1-20 minutes. The gradual transitions in the dataset include dissolve, fade in fade out, and sliding in sliding out.

5.5.2 Settings

We use the inference mode of TransNet V2 in a Tensorflow 2.0 environment to perform shot boundary detection to the input videos as a first step of their analysis.

5.5.3 Results

TransNet V2 network achieves SoA results to benchmark datasets, as shown in *Table 2*.

Table2: TransNet V2 evaluation to benchmark datasets (Souček & Lokoč, 2020)

	ClipShots	BBC Planet Earth	RAI
F1 Scores	77.9%	96.2%	93.9%

Bellow, we present results from shot boundary detection applied to a video analysed by the SD module in the context of xR4DRAMA's PUC2.

[9] https://github.com/soCzech/TransNetV2

[10] https://aimagelab.ing.unimore.it/imagelab/

[11] https://github.com/Tangshitao/ClipShots

Figure 8: SD module detected the different shots of the input video. The frames depicted here are characteristic of each video shot.

5.6 Scene recognition

5.6.1 Dataset description

We used two datasets for the training of the xR4DRAMA Scene Recognition (SR) model:

1. The **Places dataset**[12] that contains 1,803,460 training images with the image number per class varying from 3,068 to 5,000. The validation set has 50 images per class and the test set has 900 images per class.
2. The **SUN dataset**[13] that contains 908 scene categories with a varying number of images per category.

Following the user needs of xR4DRAMA, we selected specific classes that are depicted in *Table 3*. Classes "cathedral_outdoor", "sea_cliff", "stadium_baseball" and "stadium_football" are taken from the SUN dataset. The rest of the classes come from the Places dataset. "Indoor" class contains a compilation of indoor images (from the Places dataset), so that out network learns to discriminate between indoor and outdoor scenes.

Table 3: The 99 selected scene categories supported by the xR4DRAMA SR model.

airfield	airport_terminal	alley	amphitheater	apartment_building
archeological_excavation	army_base	athletic_field	badlands	bar
beach	beach_house	boardwalk	bridge	building_facade
cafeteria	campus	canal	castle	cathedral_outdoor
church_outdoor	cliff	coast	coffee_shop	construction_site
cottage	courthouse	courtyard	creek	crosswalk
doorway_outdoor	downtown	driveway	embassy	excavation
farm	field	field_road	fire_escape	fire_station
forest	formal_garden	gas_station	gazebo_exterior	glacier
grotto	harbor	heliport	highway	hospital

[12] http://places2.csail.mit.edu/

[13] https://groups.csail.mit.edu/vision/SUN/hierarchy.html

hotel_outdoor	house	industrial_area	inn_outdoor	lawn
lighthouse	mansion	manufactured_home	market_outdoor	mausoleum
motel	mountain	museum_outdoor	office_building	palace
park	parking_garage_outdoor	parking_lot	patio	pier
plaza	railroad_track	residential_neighborhood	restaurant	river
rock_arch	rope_bridge	ruin	sea_cliff	shopfront
skyscraper	stadium_baseball	stadium_football	staircase	street
subway_station	swimming_pool_outdoor	tower	train_station_platform	valley
viaduct	village	water_tower	waterfall	wind_farm
windmill	youth_hostel	zen_garden	indoor	

Examples of different categories from the training data are illustrated in *Figure 9*.

Figure 9: Examples of images used for training

5.6.2 **Settings**

Based on the scene recognition methodology we have described in Section 4.2, the model is pre-trained on the full Places dataset for the first 14 layers and on the selected classes from

both Places and SUN datasets for the last two layers. We trained the SR model using Keras[14] and Tensorflow[15] as backend. In total, we used 380927 images for the training and 95231 images for the validation of our SR model.

5.6.3 **Results**

For the evaluation of the SR model, we examined several combinations of parameter settings, in order to select the best performing model. The selected parameters and their results are presented in *Table 4*.

Table 4: Performance of the SR model.

Trainable Layers	Batch Size	Number of classes	Validation Accuracy Top1	Validation Accuracy Top5
2	32	99	52.46%	83.02%

Results from the SR model on xR4DRAMA's visual data in the context of PUC2 are demonstrated in *Figure 10*.

Figure 10: Results of xR4DRAMA's SR model

[14] https://keras.io/

[15] https://www.tensorflow.org/

5.7 Emergency classification

5.7.1 Dataset description

For the training of the Emergency Classification (EmC) model the following datasets have been used. The datasets contain flood and fire images, along with images with no emergency situation depicted in them, so that the EmC model is trained properly.

Table 5: Datasets used for the training of the EmC model

Dataset	Flood Images	Fire Images	Other Images
MediaEval 2017 Multimedia Satellite Task[16]	1920	-	3360
European Flood 2013 Dataset[17]	3151	-	-
Roadway Flooding Image Dataset[18]	441 (includes segmentation masks)	-	-
Bowfire[19]	-	118	107
Corsican fire database[20]	-	1135	-
Fire-Detection-Image-Dataset[21]	-	109	537
Fire-Smoke-Dataset[22]	-	1000	999
FiSmo (part of)[23]	-	1885	3583

5.7.2 Settings

As mentioned in Section 4.3, for the training of the EmC model a VGG-16 architecture pre-trained to the Places-2 dataset was used. The adaptation made were the following: i) the final Fully Connected (FC) layer was removed and replaced with a new FC layer with a width of 3 nodes freezing the weights up to the previous layer and ii) a softmax classifier was deployed so as to enable multi-class recognition. The train EmC model can recognise three types of images: “Flood”, “Fire” and “Other”.

[16] http://www.multimediaeval.org/mediaeval2017/multimediasatellite/

[17] https://github.com/cvjena/eu-flood-dataset

[18] https://www.kaggle.com/saurabhshahane/roadway-flooding-image-dataset

[19] https://bitbucket.org/gbdi/bowfire-dataset/src/master/

[20] https://github.com/cair/Fire-Detection-Image-Dataset

[21] https://github.com/cair/Fire-Detection-Image-Dataset

[22] https://github.com/DeepQuestAI/Fire-Smoke-Dataset

[23] https://github.com/mtcazzolato/dsw2017

5.7.3 **Results**

For the evaluation of the EmC model, we examined several combinations of parameter settings, in order to select the best performing model. EmC model achieves a mean accuracy recognition rate of 87.32%, as presented in *Table 6*.

Table 6: Performance of the EmC model.

Trainable Layers	Validation Accuracy	Number of classes
2	87.32%	3 ("flood", "fire", "other")

Results from EmC model on xR4DRAMA's visual data are demonstrated in *Figure 11*. We tested flood and non-flood images.

Figure 11: Results of xR4DRAMA's EmC model.

5.8 **Photorealistic style transfer**

For the implementation of the first version of photorealistic style transfer we use the keras and tensorflow implementation[24] for photorealistic style transfer that does not need any

[24] https://github.com/ptran1203/photorealistic_style_transfer

further post-processing steps for WTC^2. It is an end-to-end model that can stylise images efficiently giving photorealistic quality without any post-processing. The model is pre-trained on the provided dataset of Luan et al., (2017).

5.8.1 **Settings**

We test the accuracy of building and object localisation using the photorealistic style transfer and we test it with many different parameters. We use gamma correction on the luminance channel of each image and then we apply photorealistic style transfer on this generated data. We evaluate the localization models in images with different gamma before and after photorealistic style transfer to see how the performance of localisation varies.

5.8.2 **Results**

In this section, we provide qualitative and quantitative results in object localisation using various lighting conditions and photorealistic style transfer.

In *Figures 12 and 13*, we present some qualitative results in photorealistic style transfer module using as input the same content image with different gamma correction values in luminance channel, stylised with different styles. In the case of *Figure 12*, the content images has gamma correction values from 0.15 to 0.60 and as we observe the content image is too dark and it is difficult to see what it is illustrated. After the photorealistic style transfer application the outputs images with the most of the style images, are much better and the content is clear. In *Figure 13*, there are input content images with gamma correction values close to 1. These content images are or still dark or too brightness so the content is also deformed but the stylised outputs are close to the original content image.

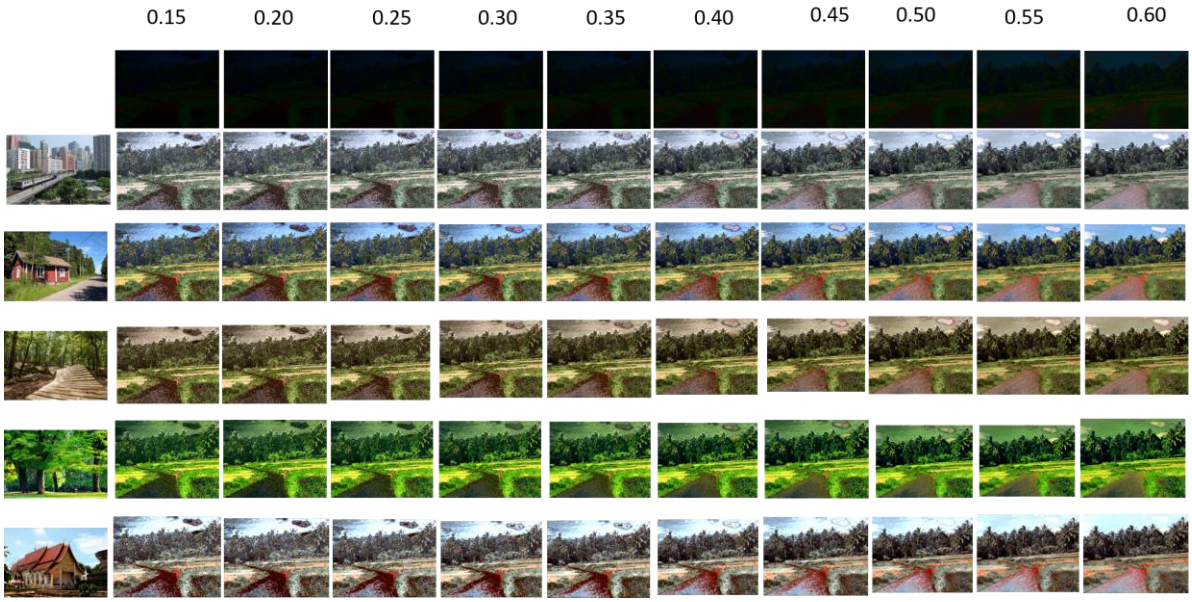

Figure 12: Photorealistic style transfer with gamma correction in luminance channels from values 0.15 to 0.6.

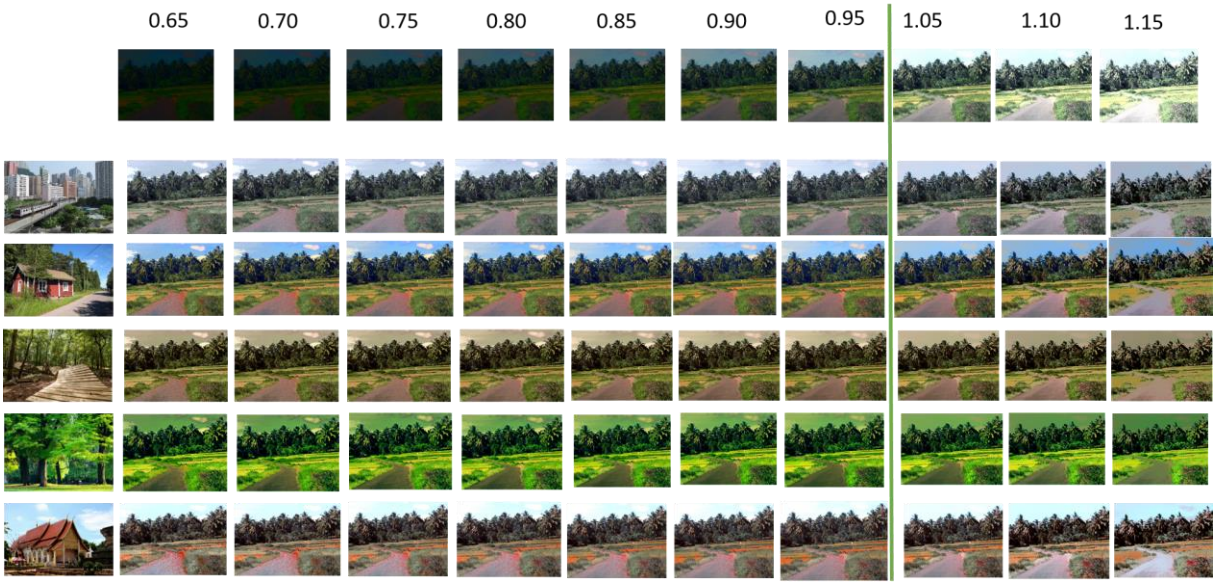

Figure 13: Photorealistic style transfer with gamma correction in luminance channels from values 0.65 to 1.15.

We use the most popular quantitative measures to measure the quality of the output images, such as MSE, RMSE, SSIM and PSNR. In *Figure 14*, we present the plots of these measures for images with different gamma corrections after photorealistic style transfer. As it is expected, the outputs with minimum MSE, RMSE and maximum PSNR and SSIM values for gamma corrections are closer to the input that corresponds to the original image with perfect lighting conditions.

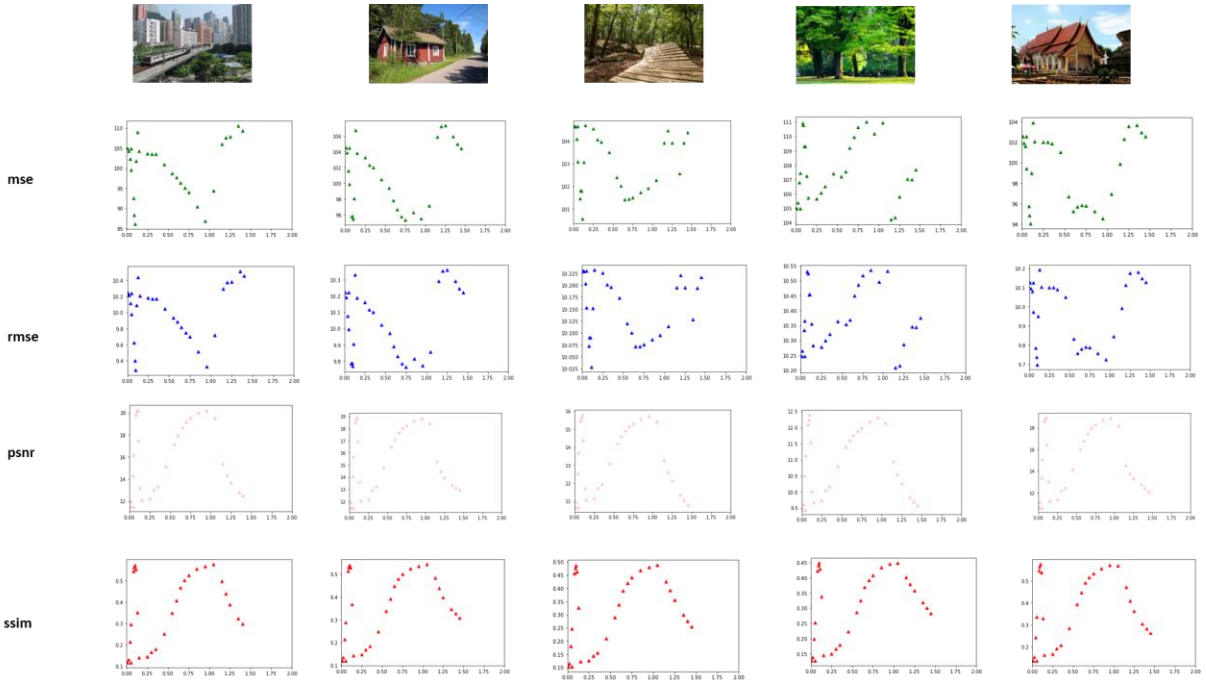

Figure 14: Plots of MSE, RMSE, SSIM and PSNR for images with different gamma corrections after photorealistic style transfer.

Finally, we present the mIoU of the pre-trained YOLOv3[25] object detector in a part of the test set of MS Coco dataset (Lin et al., 2014) and in the same test set with different gamma correction in luminance channel. Moreover, we present the mIoU of the detector in the test set with different gamma correction in luminance channel after photorealistic style transfer using as style for each image, the original content image. As we observe in *Table 7*, the application of photorealistic style transfer in luminance values from 0.4 to 0.7 leads in higher mIoU values and improves the object localisation module. For luminance values larger than 0.7, the input image tends to be identical with the original content image (that has perfect lighting conditions), the object localisation part is not significantly improved.

Table 7: mIoU results of object localization with and without photorealistic style transfer in various lighting conditions.

Gamma correction values	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8
mIoU before pst	51.24%	53.63%	56.14%	59.24%	61.42%	63.13%	64.84%	**66.21%**	**67.56%**
mIoU after pst	**54.03%**	**57.37%**	**59.57%**	**61.49%**	**62.67%**	**64.15%**	**64.92%**	65.28%	66.10%

5.9 Building and object localization

5.9.1 Dataset description

The DeepLabV3+ network was trained on the 19 classes of the CityScapes dataset[26]. The CityScapes dataset is a large-scale dataset that contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high quality pixel-level annotations of 5 000 frames in addition to a larger set of 20000 weakly annotated frames. It is intended for assessing the performance of vision algorithms for major tasks of semantic urban scene understanding for training deep neural networks.

Table 8: The 19 classes of the CityScapes dataset supported by BOL model

road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain
sky	person	rider	car	truck	bus	train	motorcycle	bicycle	

5.9.2 Settings

We trained the DeepLabV3+ network[27] on the 2975 annotated images of CityScapes dataset using Tensorflow. As a backbone network we used ResNet101. As a training method, we used Stochastic Gradient Descent (SGD) with Momentum, along with batch normalization.

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

[26] https://www.cityscapes-dataset.com/

[27] https://github.com/rishizek/tensorflow-deeplab-v3-plus

5.9.3 **Results**

In *Table 9*, we present the performance of the image semantic segmentation model for different training steps in terms of mean Intersection over Union (mIoU) and Pixel Accuracy. Moreover, in *Table 10*, we show the performance of the finally trained model per class. We can see that its performance is over 70% for 7 out of 19 classes.

Table 9: Performance of the 1st version of the BOL model

Steps	train mIoU	train Pixel Acc	val mIoU	val Pixel Acc
72912	54.12%	69.43%	51.51%	92.47%
117552	54.24%	69.51%	51.60%	92.49%
206902	54.51%	69.17%	51.68%	92.40%

Table 10: Performance of the 1st version of the BOL model per class

#	Class	train IoU	val IoU
0	**road**	**96.55%**	**96.20%**
1	**sidewalk**	**77.03%**	**72.64%**
2	**building**	**88.49%**	**88.68%**
3	wall	30.87%	20.00%
4	fence	42.14%	35.77%
5	pole	47.39%	48.58%
6	traffic light	43.77%	43.66%
7	traffic sign	57.70%	58.86%
8	**vegetation**	**88.88%**	**89.04%**
9	terrain	58.06%	45.56%
10	**sky**	**91.02%**	**88.29%**
11	**person**	**70.07%**	**63.39%**
12	rider	15.67%	19.62%
13	**car**	**89.84%**	**87.36%**
14	truck	16.47%	18.02%
15	bus	30.20%	36.17%
16	train	21.89%	8.28%
17	motorcycle	8.25%	6.11%
18	bicycle	56.18%	59.31%

Bellow, we present some results of the BOL model that we got for videos in the context of xR4DRAMA's PUC2. In Figure 15, the BOL model successfully localised the buildings and their surroundings (fence, cars, vegetation, road, sidewalk, people, traffic signs).

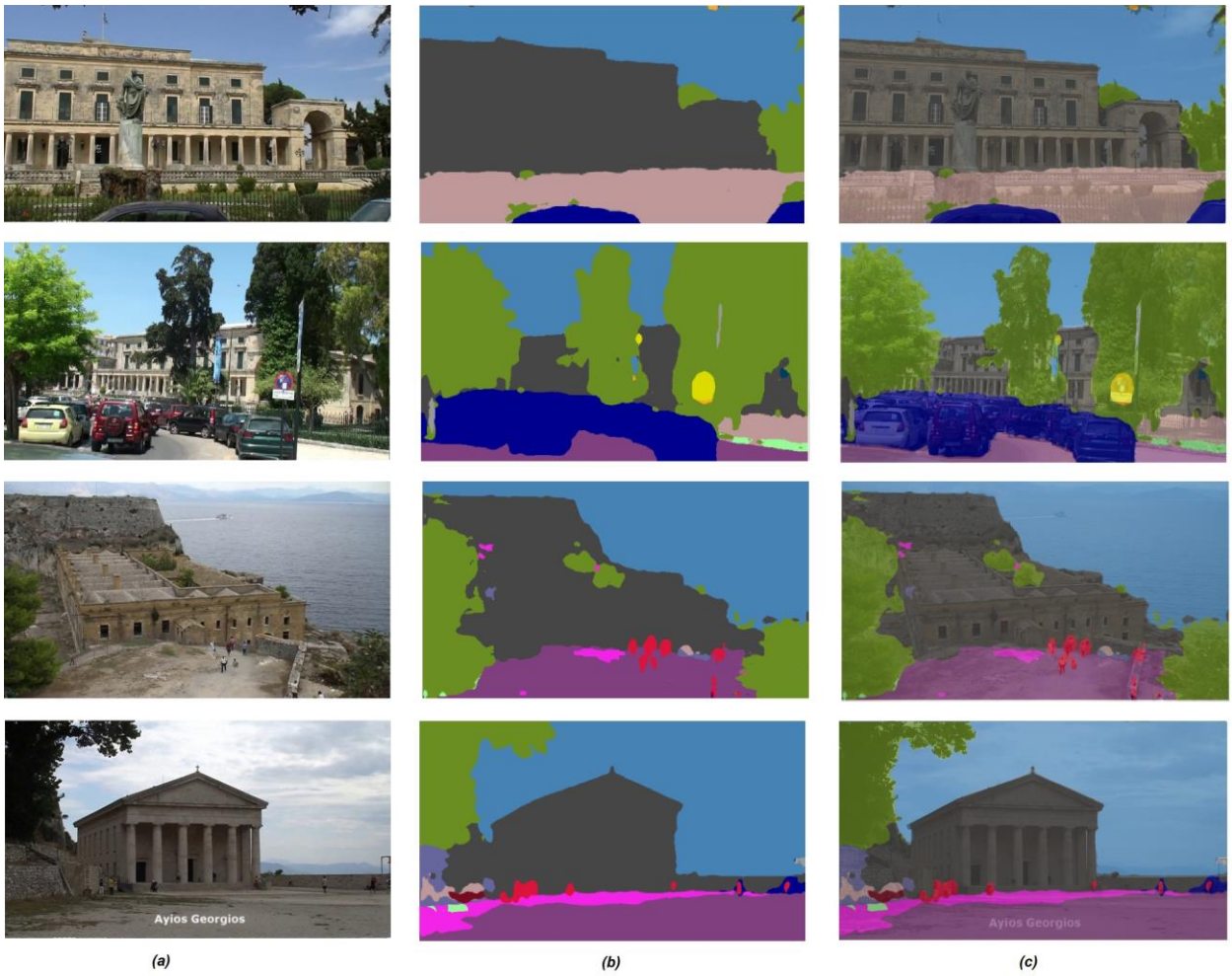

Figure 15: Successful localization of buildings and surroundings: (a) Original video frame; (b) Mask generated by the BOL model; (c) Fusion of original image and mask.

In addition, we present some results from the EmL model for images analysed in the context of xR4DRAMA's PUC1. The model localizes part of the flood water.

Figure 16: Results of the EmL model

Demo videos of the xR4DRAMA's visual analysis modules can be found in the following link: https://drive.google.com/drive/folders/10LWAKw4quRtq4m0Xap5jqNq_c3Kjd56e?usp=sharing

6 CONCLUSIONS AND NEXT STEPS

Taking into account the analyses and results presented in this deliverable, we can conclude that most of the objectives and goals associated with the final user requirements (D6.2), apart from PUC1-08, for which we will develop the proper algorithm during the second period of the project, have been successfully satisfied during the first period of the project (M1-M13). Initially, related work has been thoroughly studied and documented and then relevant datasets have been accumulated and used in order to train the appropriate scene recognition, emergency classification and building and object localization models. All relevant modules have been deployed, evaluated and tested in benchmark and xR4DRAMA visual data. Furthermore, they were also successfully integrated in the xR4DRAMA system. Technical requirements associated to each documented user requirement needs and satisfied them with the basic functionality outcome of the deployed modules.

6.1 Future work

As far as the future steps are concerned, that are envisioned to take place during the next implementation phase of T3.2, we foresee to i) develop a module for the detection of river embankment's overtopping and/or breaking to satisfy PUC1-08 (see Table 1) and ii) to make possible improvements to the already developed modules. A more precise description of the future steps for each module is given below.

6.1.1 Scene recognition (SR) and Emergency Classification (EmC)

For Scene Recognition and Emergency Classification, we envisage that we could design and deploy a sophisticated algorithm that correlates predictions from time to time, so that it can maintain the temporal coherency amongst frames. In this way, it will not produce false positive predictions for neighbour video frame intervals and will diminish the flickering classification phenomenon between sequential video frames.

6.1.2 Building and Object Localization (BOL)

As far as BOL is concerned, we envision introducing a spatio-temporal coherency for the localised buildings and objects, so that we can monitor them throughout time. This could occur by deploying a spatio-temporal tracking of the segmentation masks. In this way, it will be able to produce accurate and coherent pixel predictions for the classes that it recognizes from frame to frame. The information will give the capability to 3D reconstruction to produce accurate and reliable labels to its computed 3D models.

In addition, we plan on extensively experimenting with Photorealistic Style Transfer (PST) module in combination with BOL module, so that we enhance the performance of the latter for challenging images of poor lighting and weather conditions.

7 REFERENCES

An, J. e. (2020). Ultrafast photorealistic style transfer via neural architecture search. *Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. .*

Andrew, G. H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861.*

Baraldi, L., Grana, C., & Cucchiara, R. (2015). Shot and Scene Detection via Hierarchical Clustering for Re-using Broadcast Video. *Computer Analysis of Images and Patterns, George Azzopardi and Nicolai Petkov (Eds.). Springer International Publishing, Cham*, 801-811.

Chen, L.-C. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062.*

Chen, L.-C. a. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *pattern analysis and machine intelligence*, 834--848.

Chen, L.-C. e. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587 .*

Gangopadhyay, A. a. (2016). Dynamic scene classification using convolutional neural networks. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (pp. 1255--1259). IEEE.

Gatys, L. A. (2017). Controlling perceptual factors in neural style transfer. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3985-3993).

Gong, Y. a. (2014). Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision* (pp. 392--407). Springer.

Gygli, M. (2017). Ridiculously Fast Shot Boundary Detection with Fully Convolutional Neural Networks. *arXiv:1705.08214 https://arxiv.org/abs/1705.08214.*

Haozhi, Q. (2017). Deformable convolutional networks–coco detection and segmentation challenge 2017 entry. *ICCV COCO Challenge Workshop. Vol. 15.*

Hassanien, A., Elgharib, M. A., Selim, A., Hefeeda, M., & Matusik, W. (2017). Large-scale, Fast and Accurate Shot Boundary Detection through Spatio-temporal Convolutional Neural Networks. *arXiv:1705.03281 http://arxiv.org/abs/1705.03281.*

He, K. Z. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). IEEE.

He, M. (2017). Neural color transfer between images. *arXiv preprint arXiv:1710.00756 2.*

Huang, Y. a. (2019). Long-Short-Term Features for Dynamic Scene Classification. *IEEE Transactions on Circuits and Systems for Video Technology* (pp. 1038-1047). IEEE.

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *ECCV.*

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale Video Classification with Convolutional Neural Networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Krizhevsky, A. S. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, 1097-1105.

Kurzman, L. D. (2019). Class-based styling: Real-time localized style transfer with semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. .*

Li, Y. e. (2017). Universal style transfer via feature transforms. . *arXiv preprint arXiv:1705.08086.*

Li, Y. e. (2018). A closed-form solution to photorealistic image stylization. *Proceedings of the European Conference on Computer Vision (ECCV).*

Lin, G. a. (2017). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *computer vision and pattern recognition* (pp. 1925--1934). IEEE.

Liu, W. A. (2015). Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579.*

Lokoč, J., Kovalčik, G., Souček, T., Moravec, J., & Čech, P. (2019). A Framework for Effective Known-Item Search in Video. . *Proceedings of the 27th ACM International Conference on Multimedia.* Nice, France: Association for Computing Machinery, New York, NY, USA, 1777âĂŞ1785.

Long, J. a. (2015). Fully convolutional networks for semantic segmentation. *computer vision and pattern recognition* (pp. 3431--3440). IEEE.

Luan, F., Paris, S., Shechtman, E., & Bala, K. (2017). Deep photo style transfer. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 4990-4998).

Muhammad, H., Tahir, M. A., & Rafi, M. (2021, 12). VRBagged-Net: Ensemble Based Deep Learning Model for Disaster Event Classification. *10*.

Peng, C. a. (2017). Large Kernel Matters--Improve Semantic Segmentation by Global Convolutional Network. *computer vision and pattern recognition* (pp. 4353--4361). IEEE .

Penhouët, S. a. (2019). Automated deep photo style transfer. . *arXiv preprint arXiv:1901.03915 .*

Sergey, I., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning* . PMLR.

Simonyan, K. a. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR, abs/1409.1556*. Retrieved from http://arxiv.org/abs/1409.1556

Souček, T., & Lokoč, J. (2020). TransNet V2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning Spatiotemporal Features With 3D Convolutional Networks. *The IEEE International Conference on Computer Vision (ICCV).*

Xia, X. e. (2020). Joint bilateral learning for real-time universal photorealistic style transfer. *European Conference on Computer Vision. Springer, Cham.*

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2015). SUN database: Large-scale scene recognition from abbey to zoo. *In CVPR*, (pp. 3485–3492).

Yoo, J. e. (2019). Photorealistic style transfer via wavelet transforms. *Proceedings of the IEEE/CVF International Conference on Computer Vision.*

Zhou, B. L. (2017). Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 1452--1464.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, (pp. 487-495).