IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 12, 2021, accepted January 22, 2021, date of publication January 29, 2021, date of current version February 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3055647

COTS: A Multipurpose RGB-D Dataset for Saliency and Image Manipulation Applications

DYLAN SEYCHELL[1], (Senior Member, IEEE),
CARL JAMES DEBONO[1], (Senior Member, IEEE),
MARK BUGEJA[2], (Graduate Student Member, IEEE),
JEREMY BORG[2], (Graduate Student Member, IEEE),
AND MATTHEW SACCO[1], (Graduate Student Member, IEEE)
[1]Department of Computer and Communications Engineering, University of Malta, 2080 Msida, Malta
[2]Department of Artificial Intelligence, University of Malta, 2080 Msida, Malta

Corresponding author: Dylan Seychell (dylan.seychell@ieee.org)

The work of Dylan Seychell was supported by the Malta Government Scholarships Scheme (MGSS) for his Ph.D. studies.

ABSTRACT Many modern computer vision systems include several modules that perform different processing operations packaged as a single pipeline architecture. This generally introduces a challenge in the evaluation process since most datasets provide evaluation data for just one of the operations. In this paper, we present an RGB-D dataset that was designed from first principles to cater for applications that involve salient object detection, segmentation, inpainting and blending techniques. This addresses a gap in the evaluation of image inpainting and blending applications that generally rely on subjective evaluation due to the lack of availability of comparative data. A set of experiments were carried out to demonstrate how the COTS dataset can be used to evaluate these different applications. This dataset includes a variety of scenes, where each scene is captured multiple times, each time adding a new object to the previous scene. This allows for a comparative analysis at pixel level in image inpainting and blending applications. Moreover, all objects were manually labeled in order to offer the possibility of salient object detection even in scenes that contain multiple objects. An online test with 1267 participants was also carried out, and this dataset also includes the click coordinates of users' selection for every image, introducing a user interaction dimension in the same RGB-D dataset. This dataset was also validated using state of the art techniques, obtaining an F_β of 0.957 in salient object detection and a mean (Intersection over Union) IoU of 0.942 in Segmentation. Results demonstrate that the COTS dataset introduces novel possibilities for the evaluation of computer vision applications.

INDEX TERMS Dataset, RGB-D, salient object detection, inpainting, blending, segmentation.

I. INTRODUCTION

Modern computer vision applications are composed of a number of pipelined modules, each carrying out specific functions. These include image salient object detection, segmentation, inpainting and blending. The evaluation of each individual module has its own characteristics and requirements. A variety of datasets are available for the specific evaluation of dedicated modules carrying out the operations mentioned above. However, a single dataset that enables the evaluation of a pipelined solution is not easily found and to the knowledge of the authors, it does not exist.

This paper presents the COTS (Common Objects of a Traveling Scientist) Dataset, a travel-themed dataset containing 120 different instances organized in a selection of scenes as explained in Section III. The selected objects were configured in different scenes specifically designed to be useful in a variety of computer vision applications. These scenes are organized into two categories. The first category contains single objects, shot with a green background. The second category, contains different instances of specific scenes with multiple objects. Every instance contains an object that was not present in the previous scene. The 8-bit depth map of the

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

TABLE 1. The number of categories in every RGB-D dataset being considered and their respective number of frames or individual images.

Dataset	Categories	Frames/Images
OBJS	51	250000
BBIR	1	100
SCAN	1	10000
OSEG	1	111
GLHY	35	50
SSEG	16	NA
COTS	*29*	*120*

scene and ground truth binary image of every object for every scene in both categories is also available in the COTS dataset.

The first category of scenes serves to evaluate algorithms that measure object saliency since it includes color images and the object ground truth, similar to common datasets used in literature, such as the MSRA10K [1]. However, the COTS dataset differs as it also provides the depth map for every object. This extra information opens the possibility for the exploration of a relationship between object saliency and its depth. Furthermore, this also provides sufficient data for the evaluation of object detection and extraction algorithms.

The second category of scenes is specifically designed to address the gap in the evaluation of inpainting applications. Inpainting or object removal applications are normally evaluated using a mean opinion score (MOS) methodology [2]. While this approach serves its purpose and has its relevance from an image quality perspective, it lacks the objective rigour that is sometimes expected out of comparative results. This dataset allows for the evaluation of such techniques by also providing an actual instance of the scene without the inpainted object. The sequential nature of the dataset also allows for the evaluation of blending techniques where a new object is introduced to the scene.

The COTS dataset includes different novel aspects. It is a multipurpose RGB-D dataset that is designed for different computer vision applications. The structured and incremental approach in which scenes are built provides groundtruth within the dataset itself for applications such as inpainting and blending that traditionally lacked this type of dataset for their evaluation. Moreover, this dataset was constructed in a structured and controlled environment that is documented in detail in this paper.

An overview of existing RGB-D datasets that showcases how they are used to evaluate segmentation, inpainting, blending and salient object detection techniques is presented in the next section. The methodology behind the construction of this dataset is presented in detail in Section III. Due to the importance of user interaction data, a section is dedicated to the online test and its methodology. Section V of this paper presents a set of experiments that demonstrate how the COTS dataset can be used to evaluate a selection of techniques ranging from saliency detection to image manipulation techniques. A conclusion follows in Section VI.

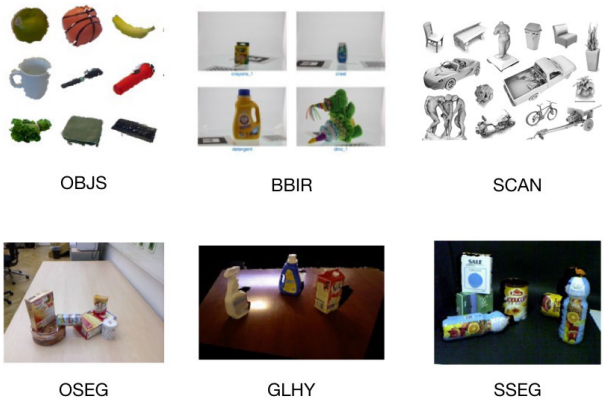

FIGURE 1. A sample from the comparable RGB-D datasets as reviewed by [3].

II. BACKGROUND

The importance of accompanying images with the respective depth information is on the rise. This is mainly due to the increase in the availability of camera technology that allows for such image acquisition. Throughout the years, a substantial number of RGB-D datasets were created. Firman [3] recently analyzed and surveyed these datasets, aggregating them into different applications. The uses for RGB-D data range across different applications, from saliency detection to object detection and classification. Another variation is the scale and nature of the environment being captured in the dataset. Some datasets, such as the COTS, are designed to focus on small objects while other datasets are developed to capture larger scenes such as a room or, more so, outdoor environments or in the wild [3]. This section focuses on RGB-D datasets that center on small objects since these are solely the ones comparable to the dataset being introduced in this paper. Moreover, datasets that have been traditionally used to benchmark saliency detection techniques will be analyzed due to the specialized application of this dataset. This paper explores the following list of RGB-D datasets:

OBJS : RGB-D Object Dataset [4]
BBIR : Bigbird Dataset [5]
SCAN: A large dataset for object scans [6]
OSEG: Object Segmentation Dataset [7]
GLHY: Global Hypothesis for Verification for 3D Object Recognition [8]
SSEG : RGB-D Semantic Segmentation Dataset [9]

A variety of different technologies can be used to capture RGB-D datasets. Before its discontinuation in October 2017, the Microsoft Kinect served as a popular tool to generate an extensive number of datasets [3]. Since its termination, the need for alternative devices for future RGB-D datasets ensued. In this context, we use the Intel RealSense D435 Depth Camera[1] to develop this dataset. In other datasets that focused on a more holistic 3D reconstruction of the

[1]The full documentation of the Intel RealSense Camera can be found on: https://realsense.intel.com

TABLE 2. A summary of the different information and data sources available within each dataset. This includes the availability of object masks, setup information, semantic segmentation data and point-clouds. This table also lists which datasets were shot within controlled lighting conditions.

Dataset	Masks	Setup Information	User Interaction	Semantic Segmentation	Controlled Lighting	Point-cloud/ 3D Mesh
OBJS	Yes	No	No	Yes	No	No
BBIR	Yes	Yes	No	No	Yes	Yes
SCAN	No	Yes	No	No	No	Yes
OSEG	Yes	No	No	Yes	No	Yes
GLHY	No	No	No	Yes	Yes	Yes
SSEG	No	No	No	Yes	Yes	Yes
COTS	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*

objects being captured, a DSLR camera in conjunction with a PrimeSense Carmine was used to generate a point-cloud for the objects [4], [6].

The *RGBD Object Dataset* (OBJS) [4] was constructed in an indoor environment. It is claimed that this dataset was constructed in a controlled environment, however, the detail in the paper related to the setup is limited. This dataset consists of single objects and the RGB-D image of the room in which the objects were placed was captured. Subsequently, using a mask, the color and depth information of the objects of interest were extracted. In this approach, attributes such as shadows and lighting across the dataset might not be preserved causing variations and inconsistencies. This was taken into consideration during the construction of the COTS dataset and defined as a main objective. The detailed process can be found in Section III.

The *Bigbird Dataset* (BBIR) [5] was developed following a very strict and structured process. This process requires that the objects are placed onto a turntable including also a calibration check-board. Three pairs of DSLR cameras with a corresponding Carmine 1.09 sensor were placed in front of the turntable. By utilizing such approach, 600 RGB-D frames were captured [5]. The Carmine sensor was also used in the *Large Dataset of Object Scans* [6]. The priority of the authors was the construction of a dataset based upon a large number of images. This was facilitated by outsourcing the acquisition process to non-professionals. For each object, video footage was obtained, and the corresponding point cloud was constructed. Consequently, the image attributes and setup across the dataset were not preserved consistently.

Other datasets [8], [9], [7] use the Kinect v1 to capture a selection of small objects that were placed on a table. The setup of these datasets follows the same line of thought of the dataset being presented in this paper. However, in these, the setup used to capture the data was not documented, and the lighting conditions in some cases might vary. This renders the quality of the dataset suitable for some tasks such as segmentation; however, the light inconsistencies make the evaluation of image manipulation methods difficult.

A similarity across all the datasets evaluated is that these have a single individual instance of static scenes. Differently, COTS was designed to incrementally include objects in a scene while leaving the previous objects in the original place and the lighting conditions static. This is explained in more detail through Figure 12.

A. SALIENCY DETECTION DATASETS

Saliency detection focuses on the detection of regions within an image that stand out more than others [10], [11]. Saliency detection can be achieved through various approaches and these vary from the original technique proposed by Itti *et al.* [12] based on the visual attention system of primates to modern deep learning approaches [13]–[16]. While Itti's approach is based on the features and visual attributes in a single image, the other deep learning approaches require extensive datasets for training. As a consequence, some datasets used saliency detection benchmarking with a large number of frames. Different datasets were designed and constructed for this purpose such as the MSRA10K [1] and the CAT2000 [17] dataset. The MSRA10K [1] dataset contains 10,000 images with their respective masks while the CAT2000 [17] dataset contains 4,000 images without a mask. Similar datasets to those mentioned here include the ECSSD dataset [18], JuddDB dataset [19] and the Pascal-S dataset [20]. These datasets can be split into two sets: the training set and the testing set. The former set is used to train the machine learning models while the latter is used to test the trained models. A common attribute amongst these datasets is that none of these contain depth information for its images. The dataset being proposed in this paper is not designed to train such machine learning models.

The COTS dataset is rather intended to pave the way towards the identification and resolution of challenges in saliency detection. The current available saliency datasets contain single objects and this becomes evident in various benchmarking exercises [10], [11], [21]. Furthermore, another challenge in saliency detection is the ranking of saliency in images that consist of more objects than one [10], [13], [22], [23].

Previous researchers tackled the challenge of saliency ranking in images at a pixel level [24], [25]. These proposed techniques attempted to rank saliency in an image based on the weight of saliency at a pixel level. The shortcoming of such an approach is when one considers the image at

object level. An object can be made up of hundreds or thousands of pixels, therefore pertaining to a single pixel is out of context in such scenario. An alternative approach to address this problem would be to split the image into segments or regions and process the weights of the segments as a whole before sorting the regions by their level of saliency [26]. Another technique that achieves similar results involves the use of deep-learning [13], [22].

The current popular saliency detection datasets do not contain depth information. It could therefore be concluded that the study of saliency detection algorithms in relation to RGB-D content is seen as a current challenge for this particular area [13], [21], [23] and the dataset being proposed in this paper aims to explore this further.

B. INPAINTING EVALUATION

The most common approaches to evaluate image manipulation techniques are subjective and based on user feedback. Inpainting techniques [27] can be evaluated by making use of the Mean Objective Score (MOS) [2] technique. In other circumstances [28], the outcome and results drawn from the inpainting techniques are presented without comparison but with a quantifiable conclusion. The use of full-reference metrics such as peak signal-to-noise ratio (PSNR), mean square error (MSE) or structural similarity index (SSIM) cannot be applied [29] when inpainting larger regions in an image such as in the case of inpainting entire objects. An exception to this would be if there exists an identical image of the same scene without the inpainted object for comparison. However, this is not always possible and can be difficult to achieve in certain scenes. Such an example is in scenes that have continuously changing elements such as environmental features like the sea or moving clouds. It is important to guarantee that all the parameters are consistent within the scene and this can be achieved through the use of a controlled environment. These considerations and restrictions were considered constantly throughout the design of the dataset being proposed in this paper.

III. DATASET CONSTRUCTION

This section provides an overview of the COTS dataset that is freely available on http://cotsdataset.info or https://github.com/dylanseychell/COTSDataset. The main motivation behind the design of this dataset was to evaluate the different stages of a computer vision application through the use of a single dataset.

The first part of this section contains images of single objects placed on a green surface in front of a background of the same color, while the second part contains themed scenes incorporating single to multiple objects.

The latter part of the proposed dataset contains 27 scenes that capture multiple objects, where every scene has multiple instances such as the example being shown in Figure 12. Every scene has an average of three instances, excluding instance 0, since this is common for each scene. There is a total of 88 instances organized into two sections which are explained below.

Throughout this dataset, one can analyze a traveling theme with the second part of the dataset containing objects that are organized by traveling aspects. This allows for easier semantic categorization while at the same time providing the opportunity to expand the dataset in the future. Careful consideration was also taken to include occluded objects in the scenes to be able to evaluate the techniques that are sensitive to occlusion and therefore be able to evaluate every aspect of these techniques.

Further considerations were taken to determine the choice of objects. Elements such as the material and the reflective property played an important role in the selection of objects. The chosen objects were made from specific materials such as transparent glass (shooter glass), polished glass (mug, tagine and statues of Buddha and Genisha), metal (travel-mug and Macbook), matt paper (Google Cardboard and most of the books), plastic (washing containers, headphones) and textile (Daydream VR headset, headgear and shoes). The selected objects for this dataset also vary significantly in size ranging from a small shooter class to a laptop and tagine.

A. DATA COLLECTION

To develop this dataset, a dedicated controlled environment was used. The setup is presented visually as a plan elevation in Figure 2. The recordings took place in an indoor environment without any external natural lighting. The only source of light used was that of two auxiliary LED lighting with modifiers that were targeted at the objects. Important considerations were made to ensure that the auxiliary lighting being used in the scene were not generating infra-red noise that would possibly affect negatively the quality of the captured depth map. In addition, a designated region on the surface was marked so that every object is placed within this region as this mark falls precisely within the camera's field of view. Moreover, the configuration of the scene was measured and recorded, keeping the setup constant throughout the scene capturing process.

The Intel RealSense depth camera D435 that makes uses of active IR stereo technology was used to capture the images that make up this dataset. The realsense-viewer tool in the official SDK was first used to calibrate the camera setting and afterwards it was used for the recording. Static scenes were recorded as a 6s video sequence and saved as a Robot OS (ROS) *.bag* file, containing all the raw data streams. The main reason behind this was that this gives the ability to preserve raw data, allowing this data to be exploited for depth/color alignment, depth measurements as well as providing suitable data for the hole filling algorithms. Furthermore, this provides an additional feature for the users making use of this constructed dataset as they can perform a more refined selection of the frame if this is required.

The Intel D435 was identified as the most suitable camera model for this dataset for a various number of reasons. Firstly, this model offers one of the highest resolutions with

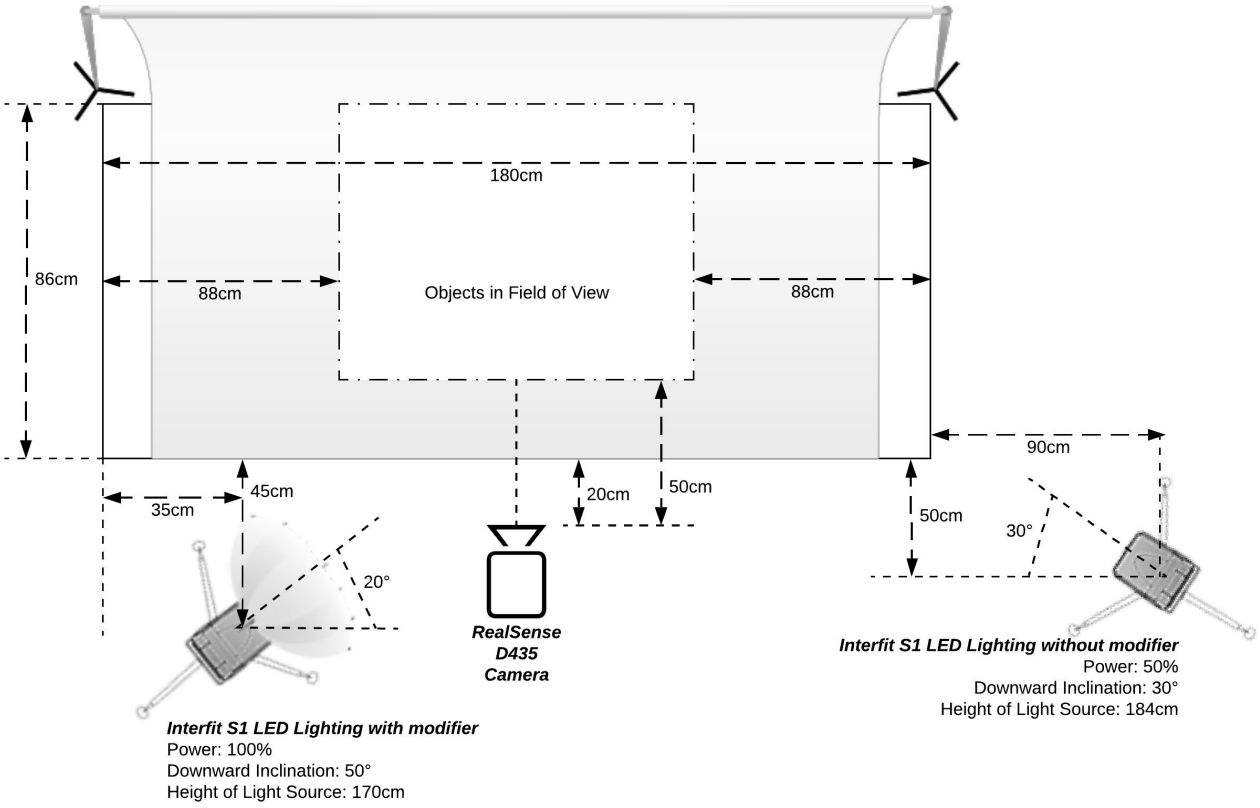

FIGURE 2. A plan elevation of the studio layout used for the data collection. This diagram is not to scale.

TABLE 3. Intel RealSense Camera Properties.

Image resolution	1280×720 pixels
Video frequency	30 Hz
Extracted scene files	Color frame (jpg), 16-bit depth frame (png), 8-bit depth frame (png), Raw ROS .bag file
Intrinsic Parameters	ppx: 623.328, ppy: 361.712, fx: 924.744, fy: 925.107
Depth scale (only used for 16-bit images)	0.001

high-accuracy depth reading within the recommended range of 0.2-7m, while being affordable. In addition, stereo cameras are usually disregarded due to their weak operation in low-texture scenes making them generally impractical for this task. However, this camera model is equipped with an active IR projector that transmits its own pattern. This feature allows for the information to be gathered even on a low-textured surface. The camera also incorporates a dedicated ASIC chip that conducts the necessary edge computing for the dense image registration problem that is required with all stereo technology in real-time. Therefore, the whole camera system outputs directly the depth information, which allows for the minimal computation on the host platform. Moreover, the Intel open-source community is one of the most informative and helpful on multiple platforms. This makes it easier to retrieve all the necessary information to get started, as well as be provided with support throughout the development. The hole-filling algorithm is introduced in the next section. This algorithm is built-in the D435 SDK and updated regularly.

1) HOLE-FILLING PROCESS

Throughout the extraction of the aligned depth frames from the recording, it could be noted that the depth map contained ''holes''. These ''holes'' represent missing information. There are various reasons for this artefact in a stereo system as portrayed in [30]:

1) **Occlusions** – This happens when the left and right image do not encapsulate the same scene or object due to shadowing. Generally, the left image is used as a reference, therefore the occlusion effect is observed on the left side of objects as well as along the left edge of that same image;
2) **Low-texture** - Stereo matching is dependent on the matching of texture in the left and right images. Therefore, for texture-less surface like, for example, a flat white wall, the depth can be challenging to estimate (this is the reason why the active projector is used to generate texture);

FIGURE 3. A photograph of the setup used for the data collection process of this dataset.

3) **Multiple matches** - There might be circumstances in which during the matching process, there exists more than one block that is found to match equally the reference one. This occurs commonly when the scene incorporates a uniform periodic structure;
4) **Signal** - A lack of signal can occur if the images are under or overexposed. In this case, there is no information retrieved;
5) **Out of range** - The stereo algorithm search-range can be exceeded if the object is very close. It is required that the objects are placed farther away than the minimum distance, Z, from the camera for it to be seen.

There are some situations in which it might be better to not deal with holes since the processing might be too intensive. This is especially true if the applications require real-time processing. However, it might also be the case where the depth-enhanced output is desired. In these cases, it is considered that a "best guess" is better than no guess at all [30].

The algorithm used for this task belongs to the spatial filtering technique. This simple algorithm makes use of neighboring pixels (left or right) within a pre-defined radius to be able to fill in the blank pixel. Different researchers discussed this simple technique as a baseline for comparing new hole filling methodologies in their various research studies [31]. For the case of the D435 camera, the left neighboring pixel is taken since the left camera is the reference. There is a total of three different methods that are available when it comes to the Intel SDK:

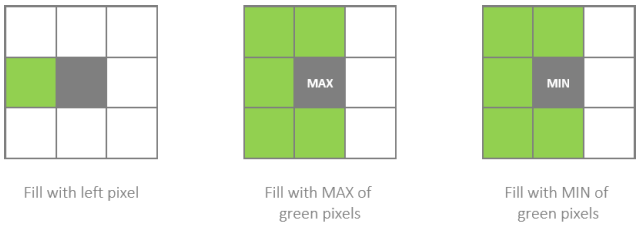

FIGURE 4. Different hole filling methods.

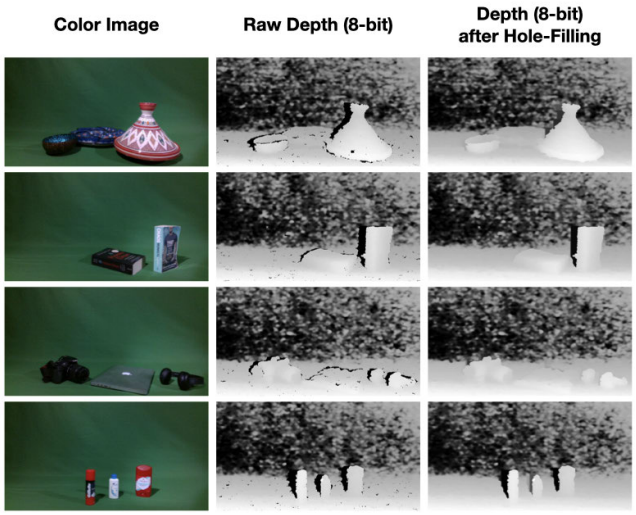

FIGURE 5. A selection of raw depth maps as captured by the camera and their result following the hole-filling process.

1) Left valid pixel value;
2) The biggest (farthest away) among the valid five upper left and down pixel values (used for depth map);
3) The smallest among the valid five upper left and down pixel values (used for disparity map).

IV. ONLINE TEST

The evaluation of saliency detection algorithms and frameworks is one of the main aims of this paper. For this aim to be reached, the color images and their corresponding depth maps and objects masks were required to be accompanied by data so that it can illustrate better how humans can relate to the dataset. This was achieved through the use of an online test that was designed to reach this objective.

This test was deployed through the use of a website as this was distinguished as the ideal platform given its potential for applications as well as its scalability. This website was shared through its URL and it was successfully conducted by 1268 participants. Its backend was specifically built for this purpose making it easy and user friendly. The users were able to efficiently view color images via an HTML page from the dataset. At the same time, the Javascript was collecting usage data in the background, making it unnoticeable to the users and hence increasing the user experience. In addition, no sensitive user data was collected in this experiment and

the usage data was stored in a hosted database. To monitor the usage activity of this online test, Google Analytics was deployed on the website and processing in the background.

During this experiment, batches of 10 images were displayed to the users, with the images shown one by one. Since the dataset contains 84 images, it provided to be a challenge to divide the dataset in such a way that would enable all the images to be presented to some user or another at a certain given point in time. The first approach considered was sequential. However, this was quickly deemed unfeasible due to the fact that this would skew the data towards the first occurring images in the dataset, since few users actually go through the entire set of 84 images. The approach that was taken to tackle this challenge was that of a tailored load-balancing algorithm that was designed and implemented in the backend to evenly distribute the images in the dataset. This algorithm requires two main components. Firstly, the images from all the dataset were required to be featured evenly and secondly that subsequent images from the same scene were not to be shown in sequence. The second requirement surfaced during the preliminary laboratory testing where it was analyzed that when a user was presented with incrementing objects of the same scene, for example as shown in Figure 12, it was likely that the user would click on the new object rather than take into consideration what is more salient in the image. Moreover, to reduce the visual bias that might impact the users performing this test, a further precaution was considered. This took the form of three carefully selected separating images that were displayed randomly before every presented dataset image. These separating images are presented in Figure 6 and these were chosen mainly because of their visual inconsistency. This was mainly implemented to reduce bias from the preceding image. This algorithm performed to its expectations, managing to successfully evenly spread all the images across the 1268 users. This resulted in every single image gathering 213 unique clicks for every image.

FIGURE 6. Separating images used between dataset images being presented to the users in order to avoid any visual bias from the previous image.

The main aim behind this experiment was to present the users with a color image from the dataset and analyze which region within the image they believe is the most salient. For each color image what was loaded individually on the screen, the users were instructed to perform the following task:

Task: *Click/Tap on the point that attracts your attention when you first see the image. The point can be anywhere and includes persons or other objects.*

The user input through the use of clicks and taps were stored for every image as coordinates. These coordinates were utilized to generate a heatmap. Furthermore, these coordinates were stored in a CSV file and are included in the proposed dataset. Moreover, during the online experiment, the movement coordinates of the mouse-enabled device were being collected. These were mainly collected when the experiment was conducted through devices such as laptops or desktop computers. Further data collected during the experiment was that of the time it took for the user to click/tap on the image after its loading. This gave further insight to the user behavior as this helped to determine if the decision was more impulsive. It was concluded that a user that clicked in a short period of time was more likely acting impulsively and therefore more likely clicking on what they considered to be more salient at first glance.

A. ONLINE TEST ARCHITECTURE

This section presents the architecture of the online data collection platform. Figure 7 presents the Data-Flow diagram of the system displaying the main modules of this architecture. There are two main data collection components; image selection (Module 1.0) and user handling (Module 1.1).

The Image Selection module (1.0) is responsible for the selection of images presented to the user as explained in Section IV. Module 1.0.1 carefully chooses the images from the COTS dataset (1.0.3) and compiles a set of images (1.0.2). This selection is then presented to the user through the HTML web page.

The User Handler module (1.1) is responsible for the presentation of the selected set of images to the user (1.1.0) and the subsequent collection of data (1.1.1). The user handling module is also responsible for storing any information about the user interaction with the images such as the cursor movements when these are available and the click coordinates. In addition, it also stores other general but related information such as the type of device being used during the experiment as well as the time of the interaction.

This rigorous architecture allowed for scalable dissemination of the online test and the successful completion of the test by the users.

B. STATISTICAL ANALYSIS

This section provides a deep evaluation and analysis of the interaction between the user and the dataset during the online experimentation. It was deduced that out of the 1268 participants that took part in this study, 77% used a smartphone while 6% made use of a tablet. The remaining participants which constitute 17% of the participants made use of a desktop or laptop computer. Therefore, since 83% of the participants were tapping the images, this would imply that no cursor movements were collected in these circumstances.

A total number of 1690 persons visited the website with 1268 participating in this study by completing this test. This implies that there was a bounce rate of only 25%.

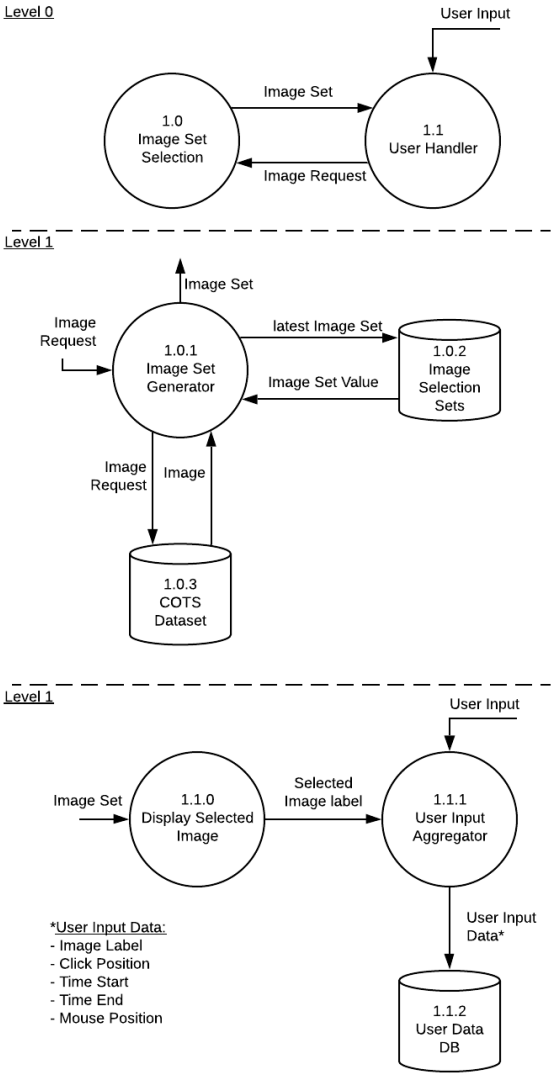

FIGURE 7. A data-flow diagram of the components in the online test.

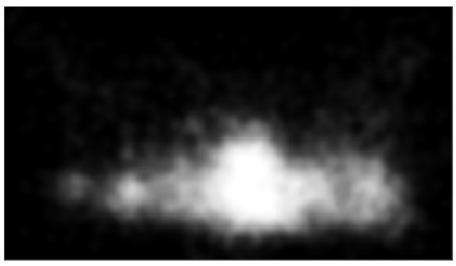

FIGURE 8. An average annotation map of all the combined clicks or taps collected from the online test.

Figure 8 illustrates the average annotation map of all the clicks and taps collected from the test. From this figure, it could be concluded that there was a very smooth overall distribution of interaction along the entire area where objects were present. In addition, it shows minimal center-bias.

It is a challenge to identify whether there were sufficient users that participated in this experiment that would enable further evaluation possible. In the research area of computer vision especially when dealing with traditional statistical techniques that calculate a minimum threshold for a population are inconsistent in such a case. This is because, throughout this experiment, we are not collecting or measuring opinion. Our evaluation focuses mainly on statistical validity which is being achieved by dividing the total number of mouse clicks into two groups with a ratio of 3:7 and then compare the distribution of x and y values between the two groups. Figure 9-a describes some of the results that were generated from the conduction of this study. Visually, the heatmap already gives sufficient information to formulate an idea about the expected result due to the clusters that focus on specific points within the image. In addition, the similarity of the x and y distribution curves across both groups emphasizes our methodology as can be seen in Figure 9-b. To further consolidate the results, we also performed a t-test that compares the x and y distribution of clicks across the two groups. Our hypothesis is defined as follows:

HypothesisH_0:

> The distribution of the clicks (x and y dimension) on the smaller sample size is similar to the distribution of the clicks on the larger sample size.

A t-test was conducted for each image evaluated through the online test and it resulted that the p-value was higher than 0.5. Therefore, the null hypothesis cannot be rejected. This implies that there was no considerable difference between the two distribution, concluding that the click/tap coordinates settled to specific regions.

C. ANNOTATION PROCESS

The evaluation of various computer vision techniques also requires a single-channel binary image mask that also serves as ground truth. These masks are black and white images to represent the object of interest, with the white pixels representing the object. These masks also play an important role in the evaluation of other techniques that generate a mask from depth information [27].

To generate the masks, the LabelMe tool [2] was used. The masks to annotate the dataset were generated by three volunteers. A mask was created for every object for every single scene. The third party annotators were not related to the project and therefore were not expected to make immediate use of the dataset. Moreover, to add a further level of independence, the annotation workshops were facilitated by a team member that was not responsible for the use of the annotated masks. A total of three masks for every object were collected, one for every annotator.

The subsequent step in the process was the inter-annotation agreement between the three masks. There are different techniques to choosing the final mask varying from choosing the smallest mask or the larger masks or also an average mask. After consideration, it was deduced that making use of such an approach might introduce a certain bias. In addition,

[2]http://labelme.csail.mit.edu

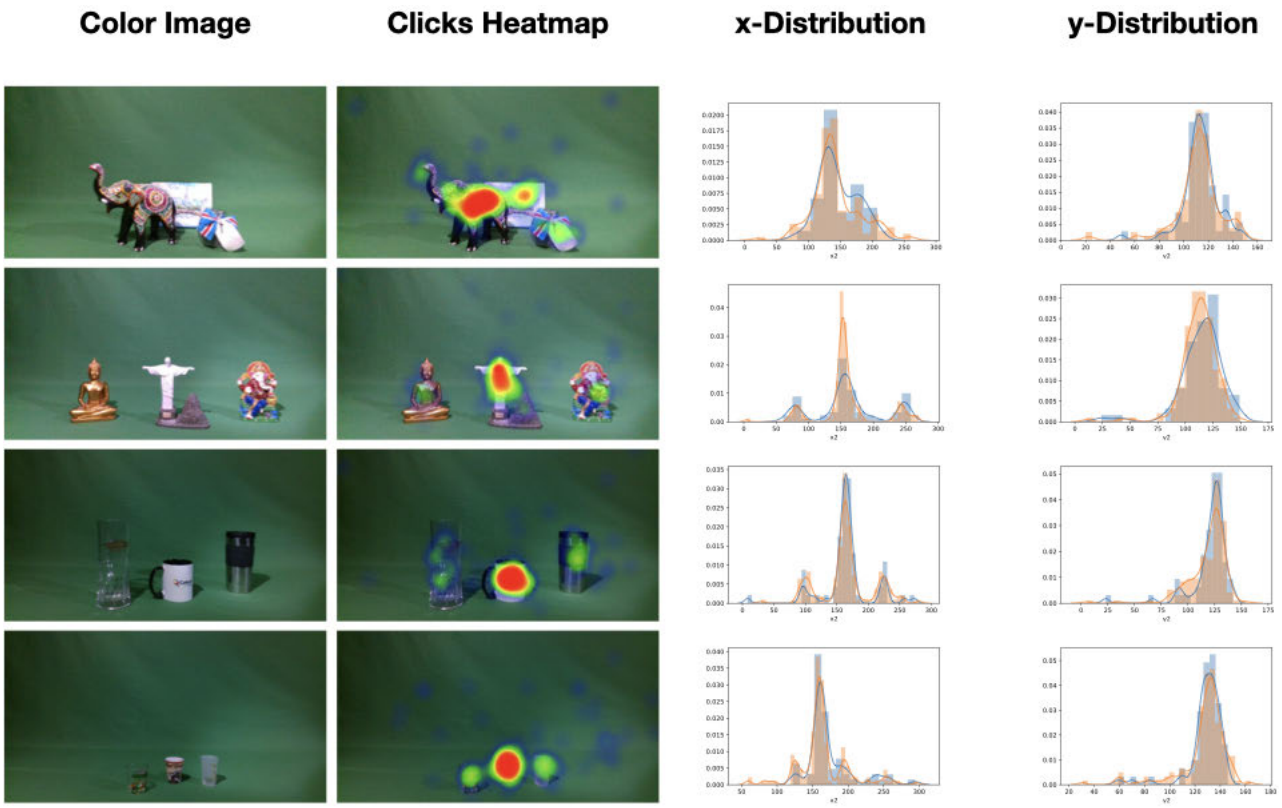

FIGURE 9. A sample of heatmaps generated from the user interaction through the online test. These are also accompanied by distribution testing of the x and y coordinates.

from experimentation, it was concluded that they were also introducing scattered white pixels in the output mask which was undesired. Therefore, it was deduced that a more conservative approach might be ideal. This approach made use of the outputs of a white pixel on the final mask only if there is a white pixel in all the three masks in the corresponding position.

1) ACCURACY OF GROUND TRUTH

The benchmarking exercise carried out in Section V demonstrates how the dataset can be used in different computer vision applications. This exercise in itself demonstrates the accuracy of the ground truth of the COTS dataset that was generated as discussed in Section IV-C.

The application of COTS in Visual Saliency presented in Section V-A. In this section it is demonstrated that the state of the art and other recently benchmarked visual saliency techniques perform as well on COTS as other saliency datasets.

Section V-B demonstrates how the COTS dataset with its groundtruth benchmarked the state of the art on the segmented output with an average of different Mean IoUs of 0.850.

The COTS dataset and its groudtruth can also be used for Inpainting and Blending as demonstrated in Sections V-C and V-D respectively. In both cases, we show how the MSE for each of these applications returns a very low result and therefore a higher quality result.

V. USAGE OF DATASET

This section demonstrates the general usage of the COTS dataset and is organized into five sub-sections. The first one explains how this dataset can be used in saliency detection modules and it is followed by three other sub-section showing how the dataset can be used in Segmentation, Inpainting and Blending applications. This section concludes by discussing how the COTS dataset can be used to evaluate different modules of a pipelined solution.

A. VISUAL SALIENCY

The process of analyzing images using visual saliency has intrigued a number of researchers and today we have a wide selection of techniques using different approaches that generate saliency maps for given images. These techniques range from eye-fixation approaches and information theory based approaches to deep-learning approaches were used to emulate human visual attention [10], [11]. Saliency techniques can be organized into two categories namely Salient Object Detection and Fixation Prediction. The process of salient object detection starts by detecting the salient objects in a scene and subsequently segmenting objects [11]. On the other hand, other models predict human fixation on a given image.

The proposed dataset can be used for the study of saliency together with the evaluation of saliency-based techniques. The first part of the dataset primarily contains images of

TABLE 4. The results of a variety of saliency models on the COTS dataset together with the MSRA10K and ECSSD benchmarking datasets, extending the work of Borji *et al.* [11]. The current state of the art technique, the Pyramid Feature Attention Network (PFAN) [42] was also evaluated on the COTS dataset with comparative results presented in the last row.

	COTS		ECSSD		PASCAL-S		DUT-OMRON		MSRA10K	
	Fixed	IDAT	Fixed	IDAT	Fixed	IDAT	Fixed	IDAT	Fixed	IDAT
			Results from [11]							
FES [38]	0.812	0.692	0.655	0.645	0.619	0.605	0.520	0.555	0.717	0.753
SR [34]	0.676	0.507	0.385	0.381	0.447	0.442	0.298	0.363	0.473	0.569
SIM [35]	0.699	0.625	0.391	0.433	0.434	0.407	0.358	0.402	0.689	0.705
SWD [39]	0.785	0.702	0.624	0.549	0.577	0.523	0.478	0.506	0.498	0.585
CA [40]	0.766	0.587	0.515	0.494	0.489	0.472	0.435	0.458	0.621	0.679
COV [36]	0.628	0.541	0.641	0.677	0.589	0.604	0.486	0.579	0.667	0.755
SEG [41]	0.951	0.941	0.568	0.408	0.534	0.344	0.516	0.450	0.697	0.585
SeR [37]	0.722	0.488	0.419	0.391	0.433	0.406	0.385	0.411	0.542	0.607
			Results from [42]							
PFAN [42]	0.957	0.842	0.931	N/A	0.892	N/A	0.856	N/A	N/A	N/A

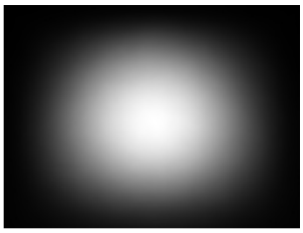

FIGURE 10. The average annotation map of the MSRA10K [1] dataset shows that the salient objects in this dataset are biased towards its center. This bias was also considered in the design of the COTS dataset and as demonstrated in Figure 8, this was mitigated accordingly by spreading salient objects further within the scene.

single objects together with the respective ground truth binary-image, similar to the MSRA10K [1] and ECSSD [18] datasets. Furthermore, the first instance of the second part of the dataset can also be used for this purpose since it also includes a single object together with the ground truth image, hence extending the number of images containing a single object within the dataset.

The COTS dataset also includes 8-bit and 16-bit depth maps for each of the objects and scenes and therefore allows for the study of the relationship between saliency and depth information such as [32]. Other prominent datasets such as the Pascal-S [20], MSRA10K [1], ECSSD [18], JuddDB [19] and DUT-OMRON [33] that are commonly used in the benchmarking and evaluation of saliency techniques do not contain any depth information for their images. The COTS dataset can therefore provide this additional value to research in this field.

1) BENCHMARKING SALIENCY

The COTS dataset was also benchmarked with other saliency models based on the methodology and source code provided by Borji *et al.* [11] where 41 saliency models were benchmarked extensively. This experiment shows that the COTS dataset can also be used to benchmark saliency models. For our experiment that extends the original study [11], 7 models were selected based on the availability of the source-code within the work of Borji *et al.* [11]. This selection included a mix of Fixation Prediction (SR [34] SIM [35] COV [36] SeR [37]) and Salient Object Detection models (FES [38] SWD [39] CA [40] SEG [41]). The COTS dataset was also benchmarked against other datasets using the state of the art technique Pyramid Feature Attention Network (PFAN) [42]. The PFAN technique scored an F_β of 0.957 on the COTS dataset as presented in Table 4 and a reported 0.931 on the ECSSD dataset [42].

The MATLAB source-code was used to carry out the comparative analysis of these 7 models on COTS together with the MSRA10K [1] and ECSSD [18] datasets. The first part of the experiment reproduced the results obtained by these 7 models on the MSRA10K [1] and ECSSD [18] datasets as found in the original study and reported in Table 4. The comparison focused on the F_β statistic based on both Fixed and Adaptive Threshold (AdpT) thresholds. The same β value of 0.3 was used to give a priority to precision over recall.

$$F_\beta = \frac{\left(1+\beta^2\right) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{1}$$

Once that the performance of the models was confirmed in the first part of the experiment, the second part of the experiment introduced the COTS dataset. The same 7 models were evaluated on the COTS datasest. Once again, both fixed and image-dependent adaptive threshold techniques were used, however this time a larger set of algorithms was tested. The F_β statistic for FES [38], SR [34], SIM [35], SWD [39], CA [40], COV [36], SEG [41] and SeR [37] was calculated. The results presented in Table 4 demonstrate that the COTS dataset can be used for the benchmarking of saliency models in the same way as other datasets. Moreover, this provides further prospects for research since the COTS dataset also includes a corresponding depth map for every image.

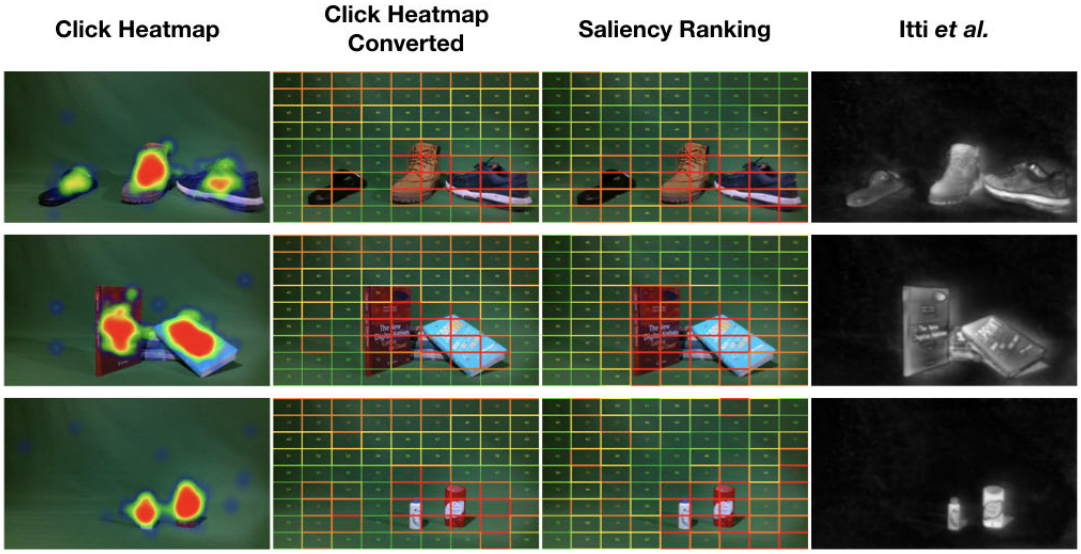

FIGURE 11. An example of how the COTS dataset can be used to evaluate saliency ranking techniques. The first column presents a visual illustration of the heatmaps collected through the online test. The second column shows how the clicks can be represented in a grid layout so that they can be compared to a saliency ranking technique [26]. The last column shows the same frame from the dataset being processed with Itti's saliency detection technique [12].

2) SALIENCY AND MULTIPLE OBJECTS

An important feature of this dataset is the information related to the user interaction in relation to every image as collected through the online test described in Section IV. The procedure outlined in [26] is followed and the segments are organized in a 9×9 grid and the following approach is used to convert the click coordinates to the desired grid segment that makes comparison possible. Every click coordinate (x, y) had to be mapped with a grid segment with index (i, j). However, each image has its own width w and height h pixels so the area covered by each grid varies according to the image size. This follows that x falls in the range $x = [0, w)$ and y in the range $y = [0, h)$ [26]. The segment $S(i, j)$ follows Equation (2) where D is the Segment Dimension, that gives the index of the respective cell as presented in Equation (3). This data allows the COTS dataset to be used in the evaluation of fixation prediction and saliency ranking techniques as demonstrated in Figure 11. Saliency ranking is in itself an emerging topic in computer vision and this dataset can be used to enable such benchmarking as demonstrated in these sections.

$$S(i,j) = \left\{ (x,y) \;\middle|\; \begin{array}{r} i\frac{w}{D} \leq x < (i+1)\frac{w}{D}, \\ j\frac{h}{D} \leq y < (j+1)\frac{h}{D} \end{array} \right\} \tag{2}$$

$$i = \left\lfloor \frac{xD}{w} \right\rfloor, \; j = \left\lfloor \frac{yD}{h} \right\rfloor \tag{3}$$

B. SEGMENTATION

The COTS dataset can also be used to evaluate segmentation techniques that make use of color and depth information such as the work presented in [43].

Segmentation is a computer vision task that extracts segments from an image that contain meaning. There are two type of segmentation tasks, semantic and instance. In semantic segmentation objects of the same type are extracted as a group of pixels. This is a useful feature that is applied in various fields [44], [45]. On the other hand, instance segmentation is thought of as a harder task and is divided into two parts. Initially an object detection process identifies individual objects in a scene/image. This process is then followed by a precise extraction of the instance of the object. Therefore, unlike semantic segmentation, in instance segmentation each object is extracted separately. Instance segmentation is also used in a number of different fields and applications [46]–[48], typically tasks that require further a distinction between objects of the same class. Recently, there has been a lot of different advances in the field of instance segmentation and one of the most popular approaches makes use of Mask R-CNN. Mask R-CNN [49] is an extension of the Faster R-CNN [50], Fast R-CNN [51] and R-CNN [52]. It is a two-stage detector, where it first extracts the Regions of Interest (ROI) by scanning an image and generating image proposals. These proposals/ROI are areas of the image that will most probably contain an object. In the second stage, the model evaluates each proposal and generates the label. The proposals are amalgamated and the bounding boxes and masks are generated.

The COTS dataset provides an alternative testing set to measure the effectiveness of the evaluators instance segmentation model. Through the masks provided with COTS the evaluator can run the segmentation model on the COTS dataset and then use evaluation metrics such as mean Intersection Over Union (IoU) or mean Average Precision (mAP).

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

where A is the ground truth and B is the prediction. The Mean IoU across the range of classes N is therefore given by:

$$MeanIoU = \frac{1}{N}\sum_{1}^{N} J(A, B) \tag{5}$$

This section outlines the procedure used in order to confirm the validity of using the COTS dataset as an alternative evaluation data-set for segmentation tasks. The Mask R-CNN instance segmentation model was chosen as it has been used extensively in the area of instance segmentation and it is considered as a baseline model for any instance segmentation task. Furthermore, an instance segmentation approach rather than a semantic approach was chosen based upon the masks available with the COTS data-set. A semantic segmentation exercise can also be computed by simply applying a morphological AND operation on the masks. Given the availability of a mask for each object it made more sense to go for the more complex approach. In terms of evaluation, the mean IoU, given in Equation 5, was used. Also known as Jaccard's Index, the IoU presented in Equation 4, is a widely used metric in instance segmentation tasks [53] where the IoU for each object is initially calculated and then the mean for each detected class is computed to output the mean IoU value. The model was pre-trained on the COCO [54] dataset, the COCO dataset contains a number of labels in common with COTS.

A number of images that had object labels in common with the COCO data-set where initially chosen. A Mask-RCNN that is trained on the COCO data-set using a ResNet-50 backbone was then used to process the pre-chosen images. These included scenes with mugs, cups and shooter glasses, food items such as vases and bowls, footwear and books. Table 5 illustrates the results obtained. For each image, the mask generated by the segmentation model was extracted. This mask was then used to generate the IoU vis-a-vis the ground-truth mask. This score value was then computed for each object present in the image. Finally, the mean IoU was calculated through the scores for each class. This task showcases how the COTS data-set can be used as an external verification tool for segmentation tasks. Similarly, other segmentation techniques can be used and trained on a variety of labels present in the COTS dataset but not available in other more known datasets. The image labels used include, technological items, hats, beanies, washing items and statues.

TABLE 5. Performance of the Mask R-CNN on the COTS dataset.

Image Name	Number of Classes	IoU per Class	Mean IoU
mugs_oc	3	0.867	0.787
		0.554	
		0.940	
mugs_oc2	3	0.920	0.857
		0.707	
		0.945	
mugs_no	3	0.833	0.905
		0.929	
		0.954	
food_no	3	0.896	0.918
		0.952	
		0.907	
food_oc	3	0.842	0.874
		0.861	
		0.918	
ip_book_no	3	0.955	0.942
		0.928	
		0.944	
ip_book_oc	3	0.866	0.602
		0.939	
		0.000	
footwear_no_2	2	0.906	0.918
		0.929	

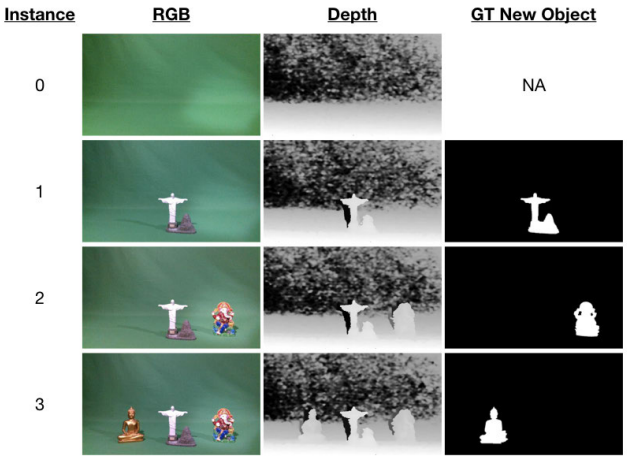

FIGURE 12. Sample of a single incremental scene for inpainting applications. The rows represent different instances of the same scene with a single object being included in every scene. For every instance, one finds the RGB image together with its respective 8-bit depth image and the ground truth binary image for the new object being included.

C. INPAINTING

Inpainting, or object removal, is the process of modifying an image such that the editing is not perceived, or its visual impact is minimized, by filling the region of interest with texture that is known from another location within the image [55]. Inpainting is achieved by either employing traditional techniques [56] or more recently by using Generative Adversarial Networks (GANs) [57].

Inpainting applications removing large objects are generally evaluated using a Mean Objective Score (MOS) such as the one specified by the ITU-T BT.500 [2]. While objective scores are helpful in the evaluation of user perception, they do not provide a quantitative measure of the efficacy of the inpainting technique employed.

The COTS dataset addresses this limitation by design as we demonstrated in [58]. Figure 12 provides a sample of how each of the 23 instance in the second part of the dataset is split into multiple instances. For every instance, there is

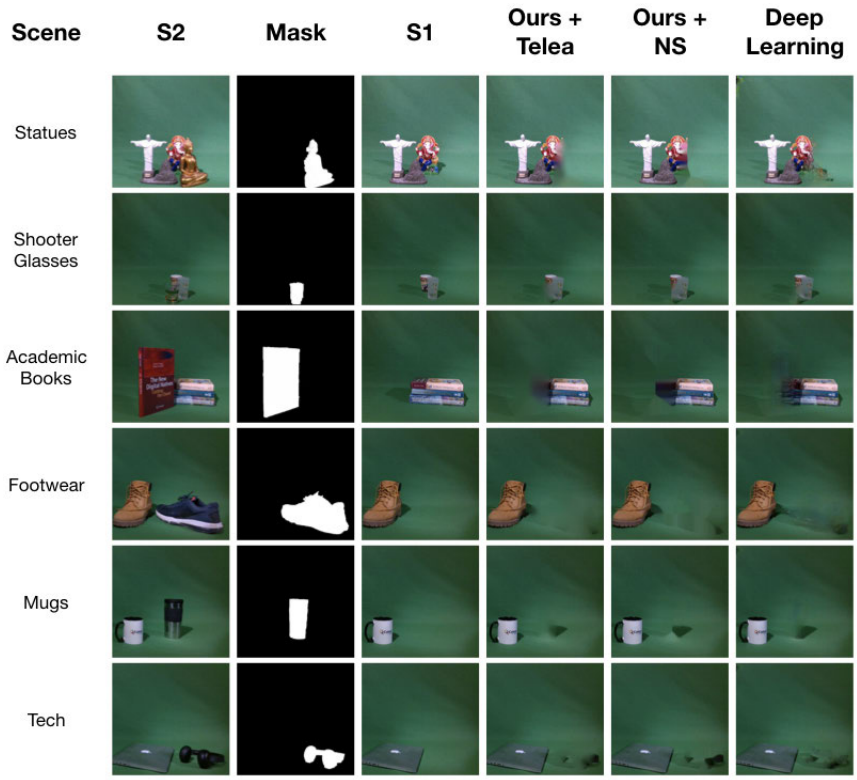

FIGURE 13. A visual representation of the comparative result of the objective evaluation inpainting techniques. S2 is the original scene upon which the inpainting is carried out using the mask presented in the second column. S1 in the third column is the actual scene without the object represented by the mask, hence acting as ground truth of the inpainting. The last three columns are the results of different inpainting techniques. These are namely our technique with Teala's [59] and Bertalmio *et al.* [60] and the NVIDIA deep learning method of [61] in the last column.

a progression in the way images are structured where an instance has an object that was not present in the one before it. Every new instance has only one new object included with nothing else in the image being modified. Such consistent progression provides the desired environment for the evaluation of inpainting techniques. Since such techniques remove an object from a specific instance (n), the instance before it ($n-1$) can be therefore used as ground truth as it will be missing the inpainting object, by design.

In an effort to assist the evaluation of inpainting algorithms, the ground truth in the form of a binary image is provided for every instance with the methodology outlined in Section IV-C. Evaluators can use this binary mask to guide the inpainting algorithm under evaluation without the need of generating their own mask through segmentation techniques that might make comparison more challenging. The use of the COTS dataset in such an experimental setup is demonstrated using the framework presented in [27]. The experiment is presented in Figure 14. In this example, the target object for inpainting is the red deodorant in Scene 2 marked as S2. The object was inpainted using the Telea approach [59] within the [27] framework. Scene 1 (S1) is the same as S2 less the red deodorant. Therefore, S1 can be used as ground truth for S2 less the target object. The inpainting result is compared to S1 using a full-reference metric. In this case, the MSE was used for the comparison.

TABLE 6. The MSE metric result for each scene where the inpainting result was compared to S1. The maximum error signifies the MSE reading when S2 was compared to S1, therefore comparing the scene without the object with the scene including the target object.

		Mean Squared Error (MSE)			
	Occlusion	**Ours + Telea**	**Ours + NS**	**Deep Learning**	**Max Error**
Statues	Yes	369.10	452.39	455.79	1139.27
Shooters	Yes	57.20	68.17	72.11	83.09
Academic	Yes	384.76	488.48	484.78	1990.00
Footwear	No	58.64	69.12	124.73	1617.40
Mugs	No	79.31	101.61	108.91	407.76
Tech	No	112.46	153.91	142.79	570.52

1) INPAINTING EXPERIMENT

The COTS dataset is ideal for comparing different inpainting techniques. This is demonstrated through the experiment presented in this section that was designed to objectively compare different inpainting techniques. A selection of six scenes with different target objects were used and the results

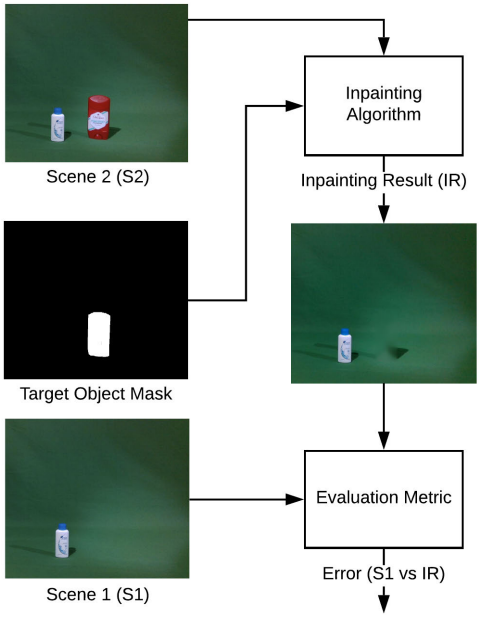

FIGURE 14. Usage of the COTS dataset and the proposed approached to evaluate an inpainting technique presented in [27].

are presented in Table 6. These scenes were split into two groups. One group contains occluded objects while the other does not. The set with occluded objects contains the following scenes: shooter glasses, statues, and academic books. The other group contains the mugs, tech and footwear scenes and none of these had any occlusion [58].

The inpainting evaluation method discussed above was followed for each of the six scenes. The scenes marked as S2 contain an object that is represented by a binary mask. S1 is an actual instance of the scene without the object and this instance was a specific feature of the COTS dataset. This framework was used to demonstrate how the COTS dataset can be used to evaluate three inpainting approaches. Two inpainting techniques used in this experiment are the ones presented in [27] first with Teala's [59] method and then with Bertalmio *et al.* [60] method. The COTS dataset was also used to evaluate a generative deep learning inpainting approach. For this part of the experiment, NVIDIA's approach by Liu *et al.* [61] was used accordingly.

The visual quality of the result in the situation where objects were placed in front of a plain background was very interesting. The nature of the plain green background of the COTS dataset exposes different strengths and weaknesses of the inpainting techniques under evaluation. Traditional dispersion based methods returned results that were blurry and this matched what was already reported in previous work [27]. On the other hand, the deep learning approach gave a more crisp result when objects were occluded even though the quality of inpainting would still not score high marks in the subjective context. Moreover, the quality of inpainting when the target object has a plain background had a visual quality comparable to the result of the traditional techniques [58].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(S1_i - IR_i)^2 \tag{6}$$

This experiment used the MSE, presented in Equation 6, as a full-reference metric to evaluate the quality of the inpainted result IR for each of the three chosen methods in relation to S1 that served as ground truth. A maximum error was computed for comparative results and this occurs when S1 is compared with S2. The comparison of the two true scenes with and without the target object returns the worst case scenario and puts the comparative results for each of the other approaches into context. The inpainted results are compared to S1 so their error should be as far as possible from the maximum error, hence closer to S1. In general, the Telea inpainting method performs the best when compared to the other techniques [58].

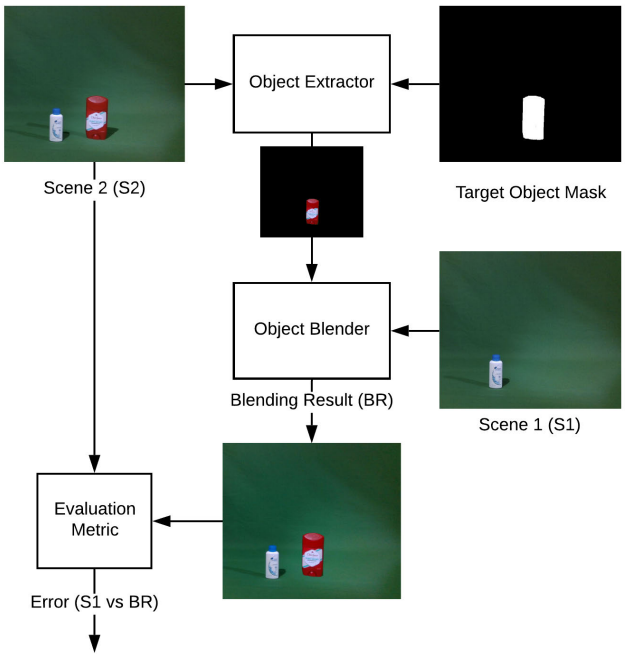

FIGURE 15. An example of how the COTS dataset can be used to evaluate blending techniques. An object is extracted from S2 and is added to S1. The blended result can then be empirically compared to S2.

D. CONTENT BLENDING

The COTS dataset can also be used to evaluate content blending or addition techniques in a similar manner to inpainting techniques. A similarly styled process is presented in Figure 15. The technique starts by identifying the target object by means of a binary image mask from S2. This time, the target object will be included into S1. The object can be identified using depth information [27] and further processing for an enhanced blending approach [62] can be used prior to blending the object into S1. The blending result is therefore S1 together with the new object extracted from S2. The position of the blended object on S1 has to be the same

as the original position of the object in S2. The result of the blending technique can also be evaluated objectively by using the MSE in a similar way to Equation 6. In this case, S2 would be used as ground truth against the Blending Result BR. Since the COTS dataset also includes shadows, one can see that while the object is blended, the shadows were not considered. Future techniques that attempt to recreate shadows can therefore also be evaluated using this same proposed dataset.

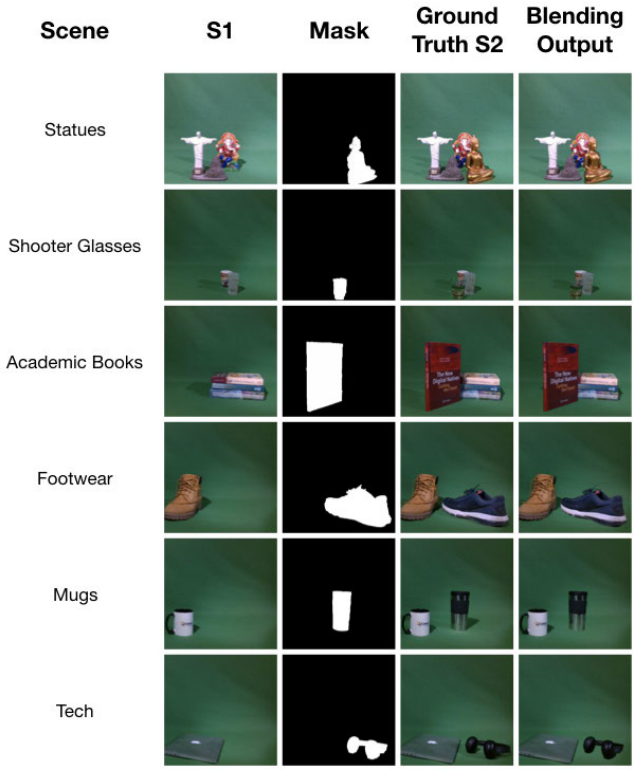

FIGURE 16. A visual representation of the evaluation of blending techniques. In this case, the additional object is taken from S2 and blended onto S1 resulting in the blending output. The ground truth in this case is S2.

An instance of this experiment using the same sample of COTS scenes as the inpainting evaluation presented in Section V-C was set up. This selection of scenes includes a set that included occluded items and another set that does not. Figure 16 presents the visual results of this experiment. The first column shows instances of S1 from different scenes that also act as the target scene where the new object will be blended. The second column presents the mask of the object that will be extracted from S2 of the same scene and blended onto S1. S2 is therefore the ground truth following the blending. The last column shows the results of the blending process. The MSE comparing the blending result to S2 is also presented in Table 7, demonstrating how COTS can be effectively used in the evaluation of blending techniques.

E. COMBINED USAGE

The sections above show how the COTS dataset can serve different types of vision applications. It nonetheless follows that the dataset can be used to evaluate different instances of pipelined solutions that use different techniques to achieve their objective. For example, the work in [27] first carries out segmentation using depth information and subsequently inpaints the segmented object. The COTS dataset can be used first to evaluate the quality of segmentation against the ground truth and then the quality of inpainting by using another instance of the same scene as explained in Section V-C.

TABLE 7. The results from the computation of the MSE of the blending technique when the blending result (BR) is compared to S2 in relation to maximum error returned when the S2 is compared to S1.

		Mean Squared Error (MSE)	
	Occlusion	**Error (BR vs S2)**	**Max Error (S1 vs S2)**
Statues	Yes	135.25	2664.17
Shooters	Yes	26.19	272.51
Academic	Yes	141.87	2177.99
Footwear	No	63.96	3444.17
Mugs	No	72.72	461.99
Tech	No	94.54	163.98

VI. CONCLUSION

This paper presented the COTS multipurpose dataset, containing 120 sets of images with their respective depth maps and ground truth. Every instance is also accompanied by a CSV file containing the click coordinates resulting from the online test in which we had an encouraging 1267 participants. This dataset can be used to evaluate a variety of computer vision applications ranging from saliency to segmentation, inpainting and blending. This offers the possibility of evaluating pipelined computer vision applications by making use of a single dataset.

This dataset also provides a number of opportunities in evaluating modern computer vision techniques and architectures. COTS is being made available for free for everyone to use as an open-source project. This current version of the COTS dataset is focused on a plain green background. This was originally intended to allow for chroma-key background replacement and therefore increase the variety and complexity of the data. Future iterations of this dataset can potentially include a set of scenes with more a more complex natural background that would increase the evaluation possibilities upon it.

ACKNOWLEDGMENT

The authors would like to thank Mr. Ryan Azzopardi, the professional photographer who assisted the team in the setting up of the studio and with the provision of lighting equipment for this setup. They also thankful to all the anonymous 1267 participants of the online test who dedicated time to go through the process and provide such precious feedback and information.

REFERENCES

[1] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[2] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, Standard ITU-T BT.500, Aug. 2012.
[3] M. Firman, "RGBD datasets: Past, present and future," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 661–673.
[4] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1817–1824.
[5] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "BigBIRD: A large-scale 3D database of object instances," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 509–516.
[6] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun, "A large dataset of object scans," 2016, *arXiv:1602.02481*. [Online]. Available: http://arxiv.org/abs/1602.02481
[7] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 4791–4796.
[8] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition," in *Computer Vision—ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 511–524.
[9] F. Tombari, L. Di Stefano, and S. Giardino, "Online learning for automatic segmentation of 3D data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 4857–4864.
[10] A. Borji, "What is a salient object? A dataset and a baseline model for salient object detection," *IEEE Trans. Image Process.*, vol. 24, no. 2, pp. 742–756, Feb. 2015.
[11] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.
[12] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.
[13] A. Siris, J. Jiao, G. K. L. Tam, X. Xie, and R. W. H. Lau, "Inferring attention shift ranks of objects for image saliency," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12133–12143.
[14] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 202–211.
[15] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang, "RGBD salient object detection via deep fusion," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2274–2285, May 2017.
[16] G. Lee, Y.-W. Tai, and J. Kim, "ELD-Net: An efficient deep learning architecture for accurate saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1599–1610, Jul. 2018.
[17] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research," in *Proc. CVPR Workshop Future Datasets*, 2015. [Online]. Available: http://saliency.mit.edu/cat2000_visualization.html
[18] J. Shi, Q. Yan, L. Xu, and J. Jia, "Hierarchical image saliency detection on extended CSSD," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 717–729, Apr. 2016, doi: 10.1109/TPAMI.2015.2465960.
[19] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2106–2113.
[20] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 280–287.
[21] R. Cong, J. Lei, H. Fu, M. Cheng, W. Lin, and Q. Huang, "Review of visual saliency detection with comprehensive information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 2941–2959, Oct. 2019.
[22] M. A. Islam, M. Kalash, and N. D. B. Bruce, "Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7142–7150.
[23] J. Zhang, F. Malmberg, and S. Sclaroff, *Visual Saliency: From Pixel-Level to Object-Level Analysis*. Cham, Switzerland: Springer, 2019. [Online]. Available: https://www.springer.com/gp/book/9783030048303
[24] L. Zhang, C. Yang, H. Lu, R. Xiang, and M.-H. Yang, "Ranking saliency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1892–1904, Sep. 2017.
[25] K. Rahul and A. K. Tiwari, "Saliency enabled compression in JPEG framework," *IET Image Process.*, vol. 12, no. 7, pp. 1142–1149, Jul. 2018.
[26] D. Seychell and C. J. Debono, "Ranking regions of visual saliency in RGB-D content," in *Proc. Int. Conf. 3D Immersion (IC3D)*, Dec. 2018, pp. 1–8.
[27] D. Seychell and C. J. Debono, "Monoscopic inpainting approach using depth information," in *Proc. 18th Medit. Electrotech. Conf. (MELECON)*, Apr. 2016, pp. 1–5.
[28] M. K. Nanduri and K. S. Venkatesh, "Segmentation directed inpainting," in *Proc. 27th Irish Signals Syst. Conf. (ISSC)*, Jun. 2016, pp. 1–6.
[29] T. T. Dang, A. Beghdadi, and M. C. Larabi, "Inpainted image quality assessment," in *Proc. Eur. Workshop Vis. Inf. Process. (EUVIP)*, Jun. 2013, pp. 76–81.
[30] G. Anders and D. Tong. *Depth Post-Processing for Intel Realsense D400 Depth Cameras*. Accessed: Apr. 20, 2019. [Online]. Available: https://www.mouser.com/pdfdocs/Intel-RealSense-Depth-PostProcess.pdf
[31] A. Atapour-Abarghouei and T. P. Breckon, "A comparative review of plausible hole filling strategies in the context of scene depth image completion," *Comput. Graph.*, vol. 72, pp. 39–58, May 2018.
[32] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *Proc. Int. Conf. Internet Multimedia Comput. Service (ICIMCS)*. New York, NY, USA: ACM, 2014, pp. 23:23–23:27, doi: 10.1145/2632856.2632866.
[33] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.
[34] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[35] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, "Saliency estimation using a non-parametric low-level vision model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 433–440.
[36] E. Erdem and A. Erdem, "Visual saliency estimation by nonlinearly integrating features using region covariances," *J. Vis.*, vol. 13, no. 4, p. 11, Mar. 2013.
[37] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, p. 15, 2009.
[38] H. R. Tavakoli, E. Rahtu, and J. Heikkilä, "Fast and efficient saliency detection using sparse sampling and kernel density estimation," in *Image Analysis*, A. Heyden and F. Kahl, Eds. Berlin, Germany: Springer, 2011, pp. 666–675.
[39] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, "Visual saliency detection by spatially weighted dissimilarity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 473–480.
[40] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2376–2383.
[41] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Germany: Springer, 2010, pp. 366–379.
[42] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3080–3089.
[43] D. Seychell and C. J. Debono, "Efficient object selection using depth and texture information," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2016, pp. 1–4.
[44] J. Jiao, Y. Wei, Z. Jie, H. Shi, R. Lau, and T. S. Huang, "Geometry-aware distillation for indoor semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2869–2878.
[45] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 578–587.
[46] T.-A. Yen, H.-C. Hsu, P. Pati, M. Gabrani, A. Foncubierta-Rodríguez, and P.-C. Chung, "NINEPINS: Nuclei instance segmentation with point annotations," 2020, *arXiv:2006.13556*. [Online]. Available: http://arxiv.org/abs/2006.13556
[47] A. Grenier, "Visual scene understanding for self-driving cars using deep learning and stereovision," Cranfield Univ., Cranfield, U.K., Tech. Rep., 2019. [Online]. Available: https://cord.cranfield.ac.uk/articles/poster/Visual_Scene_Understanding_for_Self-Driving_Cars_Using_Deep_Learning_and_Stereovision/7370174/1
[48] F. Zhang, C. Guan, J. Fang, S. Bai, R. Yang, P. H. S. Torr, and V. Prisacariu, "Instance segmentation of lidar point clouds," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Aug. 2020, vol. 4, no. 1, pp. 9448–9455.
[49] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[51] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[52] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[53] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," 2020, *arXiv:2001.05566*. [Online]. Available: http://arxiv.org/abs/2001.05566
[54] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755.
[55] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[56] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.
[57] X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," *Tsinghua Sci. Technol.*, vol. 22, no. 6, pp. 660–674, Dec. 2017.
[58] D. Seychell and C. J. Debono, "An approach for objective quality assessment of image inpainting results," in *Proc. IEEE 20th Medit. Electrotech. Conf. (MELECON)*, Jun. 2020, pp. 226–231.
[59] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, Jan. 2004.
[60] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, pp. 355–362.
[61] G. Liu, F. A. Reda, K. J. Shih, T. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 85–100.
[62] D. Seychell and C. J. Debono, "Intra-object segmentation using depth information," in *Proc. 19th IEEE Medit. Electrotech. Conf. (MELECON)*, May 2018, pp. 30–34.

DYLAN SEYCHELL (Senior Member, IEEE) received the B.Sc.IT degree (Hons.) in computer science and artificial intelligence and the M.Sc. degree in artificial intelligence from the University of Malta, Malta, in 2010 and 2011, respectively, where he is currently pursuing the Ph.D. degree in computer vision with the Department of Communications and Computer Engineering.

From 2011 to 2017, he was a Resident Academic with the Saint Martin's Institute of Higher Education, where he served as the Head of the Computing Department for five years. In 2017, he joined the Department of Artificial Intelligence, University of Malta, as an Assistant Lecturer. His research interests include visual attention, saliency, image manipulation, machine learning, and user experience design. He was awarded a number of international awards for his work, such as the Gold Seal for e-Excellence at CeBit in 2011, the First Prize by the European Satellite Navigation Competition (Living Labs), in 2010, and runner up, in 2017. In 2015, he was selected to lead the Malta's Google Developers Group. He also served as a member of the Malta Neuroscience Network and the Malta National AI Taskforce that was responsible for the development of the national AI strategy. He is involved in startups related to technology applied to heritage and tourism. He serves as a Technology Advisor and a Coordinator on a number of high-profile heritage projects.

CARL JAMES DEBONO (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from the University of Malta, Malta, in 1997, and the Ph.D. degree in electronics and computer engineering from the University of Pavia, Italy, in 2000.

From 1997 to 2001, he was a Research Engineer in the area of Integrated Circuit Design with the Department of Microelectronics, University of Malta. In 2000, he was also a Research Associate with Texas A&M University, College Station, TX, USA. In 2001, he was an Appointed Lecturer with the Department of Communications and Computer Engineering, University of Malta, where he is currently a Professor. He is also the Head of the Department of Communications and Computer Engineering and the Dean of the Faculty of Information and Communication Technology, University of Malta. His research interests include multiview video coding, resilient multimedia transmission, and wireless systems design and applications. He was a member of the management committee of the COST Action IC1105—3-D Content Creation, Coding, and Transmission Over Future Media Networks (3-DConTourNet), where he chaired the 3-D Media Coding Working Group. He is also an Editor of the IEEE Multimedia Communications Technical Committee Communications—Review.

MARK BUGEJA (Graduate Student Member, IEEE) received the B.S. degree in creative computing from the University of London, London, U.K., in 2012, and the M.S. degree in artificial intelligence from the University of Malta, Msida, Malta, in 2017, where he is currently pursuing the Ph.D. degree in artificial intelligence.

From 2013 to 2017, he worked as a Resident Academic with the Saint Martin's Institute of Higher Education, Malta. From 2017 to 2020, he was also a Research Assistant with the Department of Artificial Intelligence and the Institute of Climate Change and Sustainable Development. Since late 2020, he has been working with the Institute of Tourism Studies, Malta, and a Visiting Lecturer with the Department of Artificial Intelligence, University of Malta. His research interests include work in the area of Computer vision, reinforcement learning, and intelligent transport systems. He has a number of publications on the subjects. During this period, his research work focused on emerging technologies and Artificial Intelligence. His experience also includes various projects attributed to commercial and research interest in the area of emerging technology, such as virtual reality, augmented reality, and games.

JEREMY BORG (Graduate Student Member, IEEE) received the B.S. degree in creative computing from the University of London, in 2014, and the M.Sc. degree in artificial intelligence from the University of Malta, in 2018.

From 2014 to 2017, he worked as a Software Developer with Ixaris Systems Ltd. During this time, he specialized in quality assurance and related software development duties. Since 2017, he has been working as a Data Scientist with Global Gaming, where he is currently responsible for exploring different strategies in the commercial applications of data science. In 2018, he was selected as a Visiting-Lecturer with the St. Martin's Institute of Higher Education, Hamrun, where he also lectures Creative Computing. He also collaborates on research projects with the Department of AI, University of Malta.

MATTHEW SACCO (Graduate Student Member, IEEE) received the bachelor's degree in computer engineering and the M.Sc. degree in computer vision from the University of Malta in 2016 and 2020, respectively.

After his degree, he entered the software development industry with a local startup venturing in blockchain applications related to property management. His current research interest includes the use of RGB-D camera technology for Ambient Intelligence primarily on understanding human behavior. He aims at integrating his research into real applications varying from sport, entertainment, and assistive living.