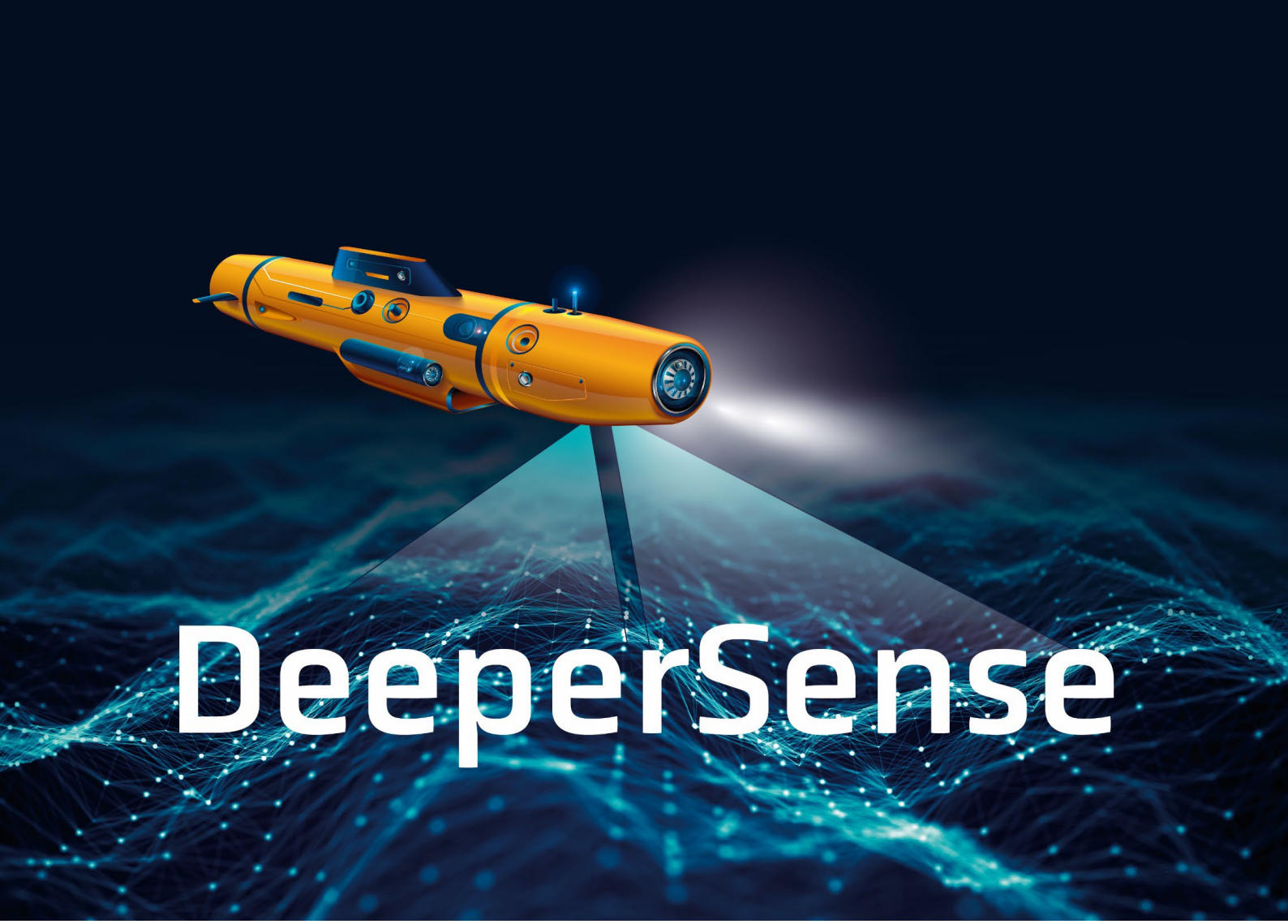


Concept ML Algorithms & Framework

Deliverable number:	2.3
Due date:	30.06.2021
Nature:	Report
Dissemination Level:	Confidential/Public
Work Package	DFKI
Lead Beneficiary:	UDG
Contributing Beneficiaries	UH
	KRA

Document History

Version	Date	Author	Description
Version 1.0	15.03.2021	Bilal Wehbe	First Draft
Version 1.1	02.06.2021	Miguel Bande Firvida	Added Relevant info
Version 1.2	25.06.2021	Bilal Wehbe	Added Framework
Version 1.3	29.06.2021	Tali Treibitz	Review
Version 1.4	30.06.2021	**Bilal Wehbe**	Final Version

Project Coordinator

Organisation: DFKI, Research Department: Robotics Innovation Center

Responsible Person: Dr. Thomas Vögele

Address: Robert-Hooke Str. 1

Phone: +49 17845 4130

e-mail: thomas.voegele@dfki.de

Consortium

Participant name	Short name	Country
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH	DFKI	Germany
Universitat de Girona	UdG	Spain
University of Haifa	UH	Israel
Kraken Robotik GmbH	KRA	Germany
Bundesministerium des Inneren	THW	Germany
Israel Nature and National Parks Protection Authority	INPA	Israel
Tecno Ambiente SL	TA	Spain

Copyright

CONTENTS

H2020-ICT-47-2020 Grant Agreement Number 101016958

List of Figures:

1. Executive Summary

The goal of this document is to identify the potential algorithms and architectures needed in the context of DeeperSense and to provide a concept of a software framework. This document is divided into 3 main sections consisting of the related work, the algorithmic concept and the software framework concept.

Related Work: An exhaustive review of the state-of-the-art literature on multi-modal machine learning methods is provided to identify a set of potential core algorithms that fulfil the needs of the use case requirements provided in D2.1 and the sensor pairing concept developed in D2.2.

Algorithmic Concept: Based on this review, a concept for a machine learning pipeline that addresses the use cases of DeeperSense is developed, detailing the scientific and technical aspects.

Framework Concept: Finally, a concept of a self-contained framework that bundles the software implementation of the machine learning methods is established.

2. Related Work

A sensor modality is defined as the means by which an instrument perceives or measures the physical world, such as sound, light, pressure, temperature, etc. A research problem is thus considered as multi-modal when two or more sensor modalities are used simultaneously to capture a scene. However, two sensors that use the same sensing method could still be considered as two separate modalities when the aspects of the sensing are different, for example an RGB and a multispectral camera are considered to be two different modalities as they use different ranges of the light spectrum. In the field of artificial intelligence, the use of data from different modalities to reason or build models to describe the physical world can be referred to as *multimodal machine learning*. Learning from multimodal sensory data can benefit to enhance the perception of the environment as well as reduce perceptual ambiguity in challenging conditions such as the ones faced underwater (Burgard, et al., 2020). Three main benefits can be identified:

- Combining multiple modalities that observe the same phenomenon may allow a deeper understanding of the phenomenon and hence, produce more robust predictions by exploring supplementary and redundant information. For example, if a plane is far away from the point of view, making it difficult to recognize, its characteristic sound will help to identify it.
- Having access to multiple modalities might allow to capture complementary information which might not be perceived by a single modality. For example, if the plane is behind a cloud and can therefore not be seen, its characteristic sounds will still make it recognizable.
- A system that fuses multiple modalities can still operate when one of the modalities is missing, and even, predict the missing modality from the existing ones. For example, by hearing the characteristic sounds of the plane, one could image how the plane might look like and where it might be located, even when the plane cannot be seen, as other planes were seen and heard in past experiences.

Given the intrinsic heterogeneity of the data, the field of multimodal learning brings some unique challenges for computational researchers. (Baltrušaitis, Ahuja, & Morency, 2018) in their taxonomy for multimodal learning identify five main challenges:

H2020-ICT-47-2020 Grant Agreement Number 101016958

Representation: Learning to represent the data coming from multiple modalities in a way that exploits the complementarity and redundancy of the different modalities. Learning representation faces many difficulties as combining data from heterogeneous sources, handling distinct levels of noise and even modality dropouts.

Translation: Learning to reconstruct a modality from one or multiple modalities perceiving the same entity. In most of the cases, there is not a unique mapping between modalities, as the relationship is open-ended or subjective, making the evaluation of the translation difficult.

Alignment: Learning to find the relationship and correspondence between elements from two or more modalities. Similar to learning translation, there may exist multiple possible alignments between elements or even indirect correspondence. Furthermore, due to the heterogeneity of the data, it is often difficult to design efficient similarity metrics to align the modalities.

Fusion: Learning to join information from two or more modalities to perform a prediction. Despite being widely researched, learning fusion still faces difficulties such as temporal unalignment between modalities. Furthermore, the modalities might also exhibit different types and levels of noise at different points in time.

Co-learning: Learning to aid the modelling of a modality with poor quality (e.g., noisy, low resolution) by exploiting knowledge and complementary information from another modality with richer quality during training. In the same way like representation, co-learning is independent task and can be always applied to better fuse, translation and/or align modalities.

For many decades, different methods have been developed and studied to address these five challenges. In recent years, deep-learning-based multimodal learning methods have attracted much attention from the research community due to their flexibility and powerful abstraction capabilities. They achieved the best results in the field so far. Newly developed technologies such as encoder-decoder models, adversarial learning, and attention mechanisms play a key role in this success.

2.1 DNN ENCODER-DECODER MODELS

Encoder-decoder models are an end-to-end architecture where the source is firstly encoded to a latent representation, which is then decoded to generate a desired output, i.e., a mapping representation of the input. This architecture can be used in several ways in multimodal learning, as it can cope with all the five challenges: Due to the multiple levels of abstraction of deep neuronal networks (DNNs), the encoder is capable of encoding one or multiple source modalities by fusing and aligning them in a joint representation (also called unimodal representation), which is then translated in the decoder to a generated modality, or existing one, which has been enhanced by co-learning.

Depending on the characteristics of the input and output modalities, different DNNs can be applied for the encoder as well as the decoder. The most popular DNNs to encode and decode images are convolutional neural networks (CNNs) (LeCun, Bengio, & Hinton, 2015). Text and acoustic signals are mainly encoded and decoded with Recurrent Neural Networks (RNNs) (Mao, et al., 2014) (Sak, Senior, Rao, & Beaufays, 2015) . Although in recent years, text has also been processed with impressive results by transformers (Vaswani, et al., 2017) (a more complex encoder-decoder architecture with attention mechanisms), and acoustic signals with CNNs (Aytar, Vondrick, & Torralba, 2016). While 2 dimensional CNNs (2DConvNet) are commonly used for encoding images, this is not the case for videos, as they exhibit a temporal consistency that cannot be encoded or

H2020-ICT-47-2020 Grant Agreement Number 101016958

modelled by 2DConvNets. To deal with the time constrains of frame sequences, 2DConvNet commonly are either combined with RNNs, such as LSTMs (Xingjian, et al., 2015), or extended with a temporal dimension (3DConvNets) (Baccouche, et al., 2011).

2.2 Generative Adversarial Network (GAN)

There are several alternative learning frameworks based on encoder-decoder models, including variational auto-encoders (VAEs) (Kingma & Welling, 2013) and generative adversarial networks (GANs) (Goodfellow, et al., 2014). Remarkably, GANs have achieved the most impressive results, by increasing the level of detail and realism, particularly of images and videos. For less than a decade, GANs have been a game changer in several fields, such as image generation (Brock, Donahue, & Simonyan, 2018) (Karras, Aila, Laine, & Lehtinen, 2017) (Karras, Laine, & Aila, 2019), cross-domain image and video translation (Isola, Zhu, Zhou, & Efros, 2017) (Zhu, et al., 2017) (Huang, Liu, Van Der Maaten, & Weinberger, 2017) (Wang, et al., 2018) (Park, Liu, Wang, & Zhu, 2019) (Wang, et al., 2018) as well as image enhancement (Karnewar & Wang, 2020) (Wang, et al., 2018) (Li, et al., 2021).

GANs are capable of learning how to model the input distribution by training two competing (and cooperating) networks referred to as generator G and discriminator D. The goal of the generator is to learn to generate fake data that fools the discriminator. Meanwhile, the discriminator is trained to distinguish between fake and real data. As the training progresses, the discriminator will hopefully no longer be able to distinguish the difference between the data synthetically generated by the generator and the real data. From there, the training process is assumed to have converged, and the discriminator can be discarded. The generator can now be used during deployment.

The main problem with the use of GANs is achieving stable training, as a faster convergence of the discriminator will lead the generator to no longer receive sufficient gradient updates for its parameters and hence, fail to converge. Various improvements to the vanilla GAN framework presented by (Goodfellow, et al., 2014) have been proposed to cope with this problem, such as CGAN (Mirza & Osindero, 2014), ACGAN (Odena, Olah, Shlens, & PMLR, 2017), WGAN (Arjovsky, Chintala, & Bottou, 2017), LSGAN (Mao, et al., 2017), BiGan (Donahue, Krähenbühl, & Darrell, 2016). Other variations have been proposed to improve other aspects as image quality, such as PatchGAN (Li, Wand, & Springer, Precomputed real-time texture synthesis with markovian generative adversarial networks, 2016), BigGAN (Brock, Donahue, & Simonyan, 2018) and ProgressiveGAN (Karras, Aila, Laine, & Lehtinen, 2017), as well as disentangled representation, such as InfoGAN (Chen, et al., 2016), StackedGAN (Huang, Liu, Van Der Maaten, & Weinberger, 2017), and StyleGAN (Karras, Laine, & Aila, 2019) - to name just a few particularly remarkable variations.

2.3 Transformers in Computer Vision

Though CNN based architectures have proven their worth for nearly a decade in solving various computer vision tasks, they come with certain inherent limitations namely viewpoint invariance and lacking a global contextual understanding. After the huge success of Transformers (Vaswani, et al., 2017) (Devlin, Chang, Lee, & Toutanova, 2018) (Radford, Narasimhan, Salimans, & Sutskever, 2018) in the field of natural language processing, a team from Google Brain (Dosovitskiy, et al., 2020) transferred these principles to computer vision tasks achieving SOTA performance by solely relying on self-attention mechanisms with significantly fewer computational costs

H2020-ICT-47-2020 Grant Agreement Number 101016958

as compared to their CNN counterparts. The architecture, termed Vision Transformer (ViT), generates a linear projection of the input image after dividing it into a sequence of flattened patches (similar to the sequence of word embeddings that the original text-based transformer operates on) and adds learnable positional embeddings to each patch (thereby enabling it to develop a structural understanding of the image) before passing it to a standard transformer encoder. The linear projection of the inputs helps leverage the use of self-attention with images without making computations intractable for realistic input sizes if each pixel were to attend to every other pixel in the image; whereas the encoder, employing multi-head attention blocks, captures global and local context at different scales enabling the network to learn more generic patterns.

However, ViT requires pre-training on a huge dataset before it can be fine-tuned on smaller goal-oriented datasets to achieve competent results. To deal with this, researchers from FAIR (Touvron, et al., 2020) came up with a knowledge distillation-based training approach where they adopt a hybrid architecture consisting of a transformer-based student network which learns from the outputs of a CNN-based teacher network, achieving performance comparable to ViT but with significantly less amount of data.

This also fits really well in the contrastive self-supervised learning model -- one can simply replace the CNN-based backbone with a Transformer architecture. DINO (Caron, et al., 2021) and MoCo-v3 (Chen, Xie, & He, 2021) are two such approaches which hold ground as the current SOTA in self-supervised learning.

2.4 Semi-Supervised Learning

Supervised deep learning approaches only work so long as they are fed with huge datasets and fail to generalize well otherwise. Areas of applications such as seabed classification, however, lack the availability of such a large number of labelled samples for training. Semi-supervised learning is a class of deep learning approaches that sits mid-way between deep supervised and unsupervised learning. They overcome the need of having huge, labelled datasets by leveraging information from the set of unlabelled data to train a network that can make better predictions than what it would have by only using the much smaller set of labelled data.

There has been a plethora of work done in this direction (Ouali, Hudelot, & Tami, 2020) such as proxy-label methods where a network is trained using the labelled set to generate “proxy” labels for the unlabelled set, a portion of which is added to the labelled set for the next iteration; or graph-based methods where the samples from the labelled and unlabelled sets are treated as nodes of the graph with the goal of propagating labels to the unlabelled nodes; or the most successful of the lot, consistency regularization methods, which revolve around the key idea that the prediction of unlabelled samples should not vary significantly when subject to realistic perturbations under the assumption that the decision boundary between two classes lies in a low-density region otherwise it is likely to slice a cluster into separate classes causing samples belonging to different classes to lie in the same cluster thereby increasing the likelihood of a sample to switch classes after a perturbation.

2.5 Self-supervised learning

Self-supervised learning (SSL) can be understood as a two-step approach of representation learning on a “pretext-task” followed by fine-tuning of the learned features on the actual “downstream” task; where the former is self-supervised in the sense that it does not require any human annotated labels but instead

generates labels from the data itself. The downstream task can be any classification, segmentation or detection problem, however, with lack of sufficient labelled data. The goal of the pretext-task then would be to serve as a (self) supervised proxy to learn meaningful embeddings of the input. In doing so, the network aims to capture enough semantics to be able solve the actual downstream task upon fine-tuning without the availability of heavy amounts of labelled samples. Pretext-tasks such as solving jigsaw puzzles (Noroozi, Favaro, & Springer, 2016) (Taleb, Lippert, Klein, & Nabi, 2021)Taleb et al. 2020], rotation prediction (Komodakis & Gidaris, 2018), image colorization (Zhang, Isola, Efros, & Springer, 2016) and even generative modelling (Pathak, Krahenbuhl, Donahue, Darrell, & Efros, 2016) (Donahue, Krähenbühl, & Darrell, 2016) have shown promising results in learning strong latent representations. In the past year, however, owing to contrastive learning, this field has seen a huge spurt of advancement with results comparable to (and even surpassing, in certain cases) SOTA supervised learning methods.

Contrastive learning revolves around the idea of constructing positive and negative pairs of the input and training a network to bring the embeddings of similar inputs closer together while pushing those of diverse inputs farther apart, essentially making the pretext-task unsupervised. Recent studies have seen different takes on the approach, each achieving SOTA-comparable results. Where SimCLR (Chen, Kornblith, Norouzi, & Hinton, 2020) necessitates the use of large batch sizes to ensure sufficient diversity in negative samples for gradient updates, MoCo (He, Fan, Wu, Xie, & Girshick, 2020) proposes a memory efficient approach of maintaining a dictionary of negative samples based on a momentum-based moving average of the main encoder to ensure consistency among encoded representations. SwAV (Caron, et al., 2020), on the other hand proposes an online clustering-based approach to learn codebook vectors which are then used for contrastive learning rather than directly comparing image features. BOYL (Grill, et al., 2020), alternatively does away with negative samples altogether and uses a bootstrapping procedure that iteratively refines its representations by using a momentum-based average of the online network as the target network for subsequent predictions. SimSiam (Chen, Kornblith, Norouzi, & Hinton, 2020) further claims that neither negative pairs nor momentum encoders nor large batch sizes are essential to avoid collapse but a stop-gradient operation and even in doing so avoids mode collapse and achieves competitive performance.

2.6 Uncertainty and interpretability of DNNs

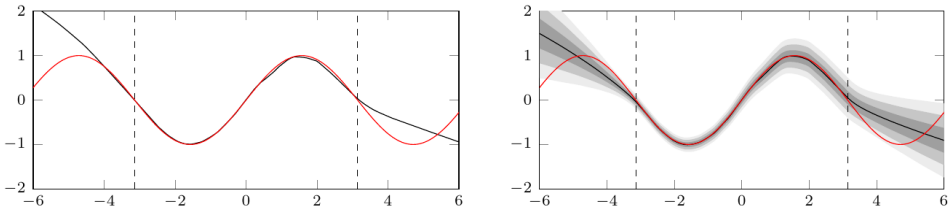
Classical machine learning models have very poor probabilistic interpretations, this means that the probabilities that a softmax classifier produces are not calibrated probabilities, and they behave abnormally as humans expect. Most models without proper uncertainty quantification produce overconfident predictions, which are the ones with high confidence/probability, but they predict an incorrect class. Humans expect that the confidence/probability associated with a prediction will correlate with the likelihood that the prediction is correct.

There are two principal kinds of uncertainty, depending on their source:

- Aleatoric Uncertainty. This is the one associated to the data, like measurement errors or stochastic processes that produce data. The fundamental property is that this kind of uncertainty cannot be reduced by adding more data or information to the learning process.

- Epistemic Uncertainty. This is the one associated to the model, like lack of training data, or inappropriate model structure. This kind of uncertainty can be reduced by incorporating more data or information to the training process.

Classical models produce a point-estimate as a prediction, without any kind of variation or uncertainty associated with it. A model with proper uncertainty quantification would produce a distribution as output, with proper aleatoric and epistemic uncertainty properties.



(a) Classic Neural Network (b) Neural Network with Output Uncertainty

Figure 1: Comparison between a classical neural network (left) and a neural network with output uncertainty

The counterpart to classical ML algorithms and neural networks is the Bayesian Neural Network (BNN), where weights are modeled as probability distributions instead of point-wise weights, and these distributions can be propagated through the network, given an input, to produce a probability distribution as the output.

The concept of uncertainty in ML also related to interpretability, as an output with associated uncertainty produces more information that is useful for the end user.

There are many methods to model uncertainty in an ML setting, below we survey a selection of them:

- Ensembles. Multiple copies of a model are trained in a dataset, and prediction variety is ensured due to random weight initialization. Predictions are combined, and the standard deviation of the output across the ensemble is used as an uncertainty metric.
- MC-Dropout/MC-DropConnect. Dropout and DropConnect are regularization methods for neural networks, that can also be activated during inference time, transforming the model into a stochastic one. Standard deviation of the output is also used as uncertainty.
- Single Model Methods. There are newer methods such as DUQ or SQR that do not require multiple models to be used, by crafting a new set of features that can model proper uncertainty. In general, the quality of their uncertainty is not known in detail.

In this project we plan to use uncertainty in ML as a way to provide additional information to the user and interpret the model's outputs.

Receptive field (RF) Visualization: There is an ongoing effort to understand the representations that are learned by the inner layers of these deep architectures. (Zhou, Khosla, Lapedriza, Oliva, & Torralba, 2014) showed that a network trained for classification can also perform detection of objects that are representative of the scene classes. They propose a data-driven approach to estimate the learned RF of each unit in each layer. They choose the top K images that activate each unit. They replicate each image many times with small random occluders (image patches of size 11×11) at different locations in the image. This results in about 5000 occluded images per original image. They then feed all the occluded images into the same network and record the change in

H2020-ICT-47-2020 Grant Agreement Number 101016958

activation as compared to using the original image. If there is a large discrepancy, then it means the given patch is important and vice versa. In this way, a discrepancy map is built for each image.

2.7 Model Compression

To reduce the demanding resource requirements of DNNs, towards real-time inference on robotic platforms, model compression methods have been extensively studied in recent years. Based on the type of strategy, five different categories for model compression can be identified (Mishra, Gupta, & Dutta, 2020):

Pruning is one of the most commonly used techniques to reduce the model size (Anwar, Hwang, & Sung, 2017) (Blalock, Ortiz, Frankle, & Guttag, 2020). Here, redundant components of the model such as channels, filters, neurons or layers that do not contribute much to increase the model performance, are removed, producing a lightweight model that consumes less power, is more memory-efficient, and provides faster interface with minimal accuracy loss.

Spare representation exploits the sparsity present in the weight matrices of the DNN model (Guo, Zhang, Zhang, & Chen, 2018). The initial idea was to remove the connexions of weights with zero or near to zero values. Furthermore, this could be extended by replacing multiple weights with alike values by a single weight with multiplex connections. If this technique is applied within a layer, it is called multiplexing, and between layers weight-sharing. In the same way than pruning, spare representation reduces storage and computational requirements with minimal accuracy loss.

Quantization is a technique in which inference is performed using arbitrary-precision integer arithmetic (Jacob, et al., 2018). By reducing the number of bits and representation complexity, arbitrary-precision integer arithmetic provides higher memory and computational efficiency than the commonly used fixed-precision floating point arithmetic. A loss of model accuracy is unavoidable by the bit reduction. However, the accuracy loss does not have to be linear when the model is trained with the low-bit representation. The model performance can even increase in some case (Mishra, Gupta, & Dutta, 2020). An extreme quantization is the complete binarization, where the floating-point operations are converted to binary operations to further reduce the storage and computational requirements of the DNN model (Hou, Yao, & Kwok, 2016).

Knowledge distillation aims to distil or transfer knowledge from a cumbersome model into a small model by training the small model in a way that achieves similar performance as the cumbersome model. In its simplified form, the non-normalized output of the original cumbersome model (teacher) serves as soft targets for training the compressed small model (student) (Hinton, Vinyals, & Dean, 2015). However, sometimes the gap between student and teacher is so large that the proper knowledge transfer is hindered. This could be prevented by inserting so-called teacher assistants between the teacher and the student, which enables a gradually knowledge distillation (Mirzadeh, et al., 2020).

3 Algorithmic Concept

3.1 SONAVision

3.1.1 Goals

H2020-ICT-47-2020 Grant Agreement Number 101016958

The main goal of the SONAVision algorithm is to learn an end-to-end association between sonar and visual camera images observing the same underwater scene. The idea is to use this learned association in order to generate realistic visual-like images, as well as depth-images when possible, given only sonar images as input or a combination of a sonar image and a dark or turbid visual image. The purpose of this algorithm is to provide images that can be easily interpreted by a human operator, even in bad visibility conditions, for example to monitor the status of a diver working in turbid or dark waters.

To further improve the perception capabilities of monitoring the diver, one of the desired features of the SONAVision algorithm is to be able to detect the diver in a given sonar image or its corresponding generated visual-like (depth)image. Furthermore, the detection of the diver could be used to estimate the diver's position relative to the sensing modalities, i.e., to the underwater vehicle, as well as the diver's body pose.

Relation to the multi-modal learning challenges:

The SONAVision algorithm as a multimodal learning method faces several challenges. The most obvious one is translation with generative models, as a clear visual-like (depth) image must be generated using a sonar image and a highly distorted visual image if present. The combination of a sonar image and a visual image requires an explicit temporal alignment and either a joint representation or direct fusion of both modalities. In order to improve the combination of the two modalities a further alignment of both point of views can be applied either explicitly, with a manual positioning of sonar and camera, or implicitly within the DNN.

Another possibility to further improve the performance of the generative model and hence, enhance the quality of the resulting (depth) image is to co-learn with a perceptual modality of higher resolution. One option could be to co-learn with a high- and low-resolution imaging sonars during training time and remove the high-resolution imaging sonars during interface time. Another option to be studied is to co-learn with the SeaVision system of Kraken Robotik GmbH, instead of a high-resolution imagining sonar, which provides dense full color 3D point cloud images with millimeter accuracy.

3.1.2 Methods:

GANs: Although GANs are mainly used to generate synthetic data, the introduction of conditional GANs (CGAN) (Mirza & Osindero, 2014), where a condition is imposed on both the generator and discriminator inputs, make them suitable for translation tasks as well. One remarkable work in cross-domain image translation that uses the principle of CGAN is pix2pix (Isola, Zhu, Zhou, & Efros, 2017). Here, the given condition is the paired image that must be transformed (see figure 1).

H2020-ICT-47-2020 Grant Agreement Number 101016958

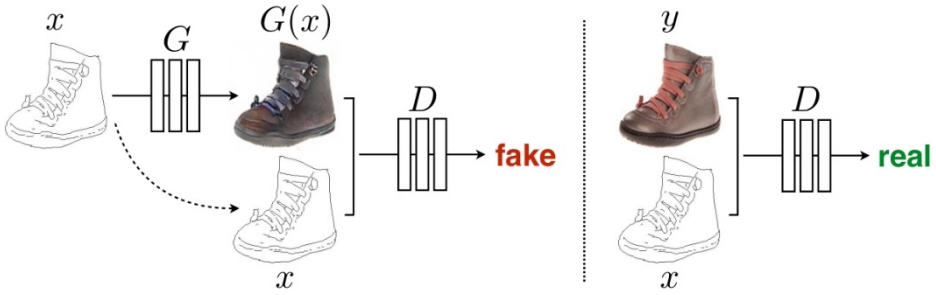


*Figure 2: pix2pix architecture schematic: a CGAN to generate translate from an edge map to a realistic photo. Unlike an unconditional GAN, both the generator **G** and discriminator **D** observe the input edge map **x** (Isola, Zhu, Zhou, & Efros, 2017).*

To keep image consistency and better model high-frequency structures during translation the authors use CGANs with the combination of U-Net [ref], i.e., skip connection between the encoder and the decoder of the generator, and PatchGAN [ref], i.e., division of the output image in N × N patches that the discriminator must classify as real or fake.

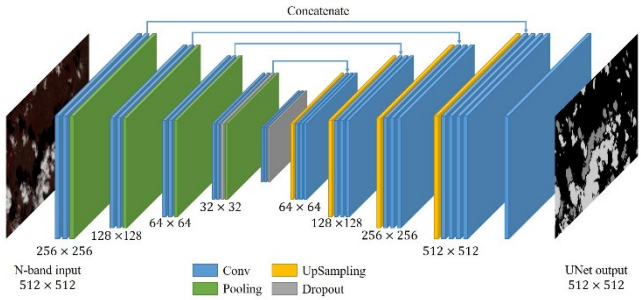


Figure 3: Example of a U-Net architecture schematic for cloud and shadow precise segmentation [Jiao et al., 2020]

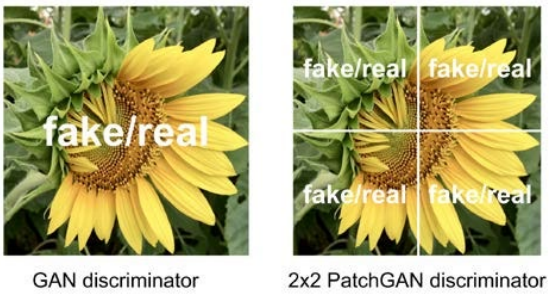


Figure 4: A comparison between GAN (left) and PatchGAN (right) discriminators (Li, Wand, & Springer, 2016)

The same architecture has been also applied in underwater domain for fish monitoring in (Terayama, Shin, Mizuno, & Tsuda, 2019), where daytime underwater images are generated from an optical underwater camera and an imaging sonar on night-time. Due to the similarity of the task and promising results, this architecture will be used as a baseline for the SONAVision algorithm.

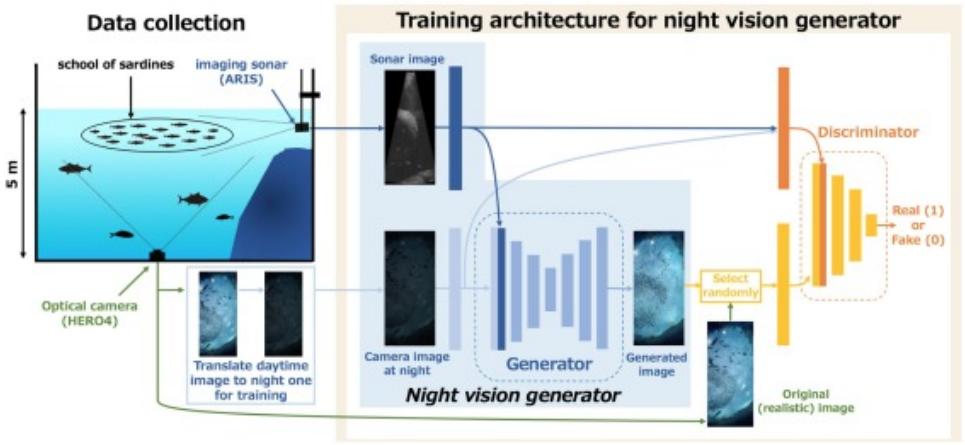


Figure 5: Schematic of the pix2pix architecture applied to generate daytime underwater images from an optical underwater camera and an imaging sonar (Terayama, Shin, Mizuno, & Tsuda, 2019)

Conditional GANs have enabled a variety of applications, but the results are often limited to low-resolution and lack of details and realistic textures, as its adversarial training might be unstable and prone to failure for high-resolution image generation tasks. Pix2pixHD [Wang et al., 2018] is an extension of the previous work, which enables the generation of high-definition images, by adding two new elements to the previous architecture: a coarse-to-fine generator, i.e., decomposition of the generator into a global generator network G1 and a local enhancer network G2, and a multi-scale discriminators, i.e., 3 discriminators that have an identical network structure but operate at different image scales.

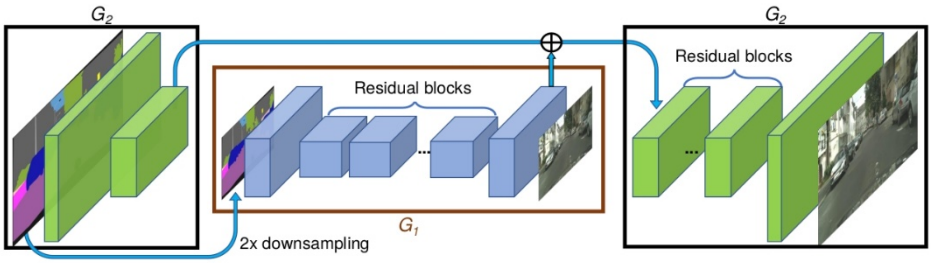


Figure 6: Schematic of the coarse-to-fine generator architecture: The residual network G_1 is trained on lower resolution in a first step. Then, the residual network G_2 is appended to G_1 and they are trained together on high resolution images (Wang, et al., 2

Another approach for the generation of high-resolution images was proposed in the same year with progressiveGANs (Karras, Aila, Laine, & Lehtinen, 2017). The authors suggest as a solution the progressive increase of spatial resolution of the generated images, by incrementally adding new layers in the generator and the discriminator after convergence. This way of training allows the learner to first discover large-scale structure of the image distribution and then shift attention to increasingly finer scale detail.

H2020-ICT-47-2020 Grant Agreement Number 101016958

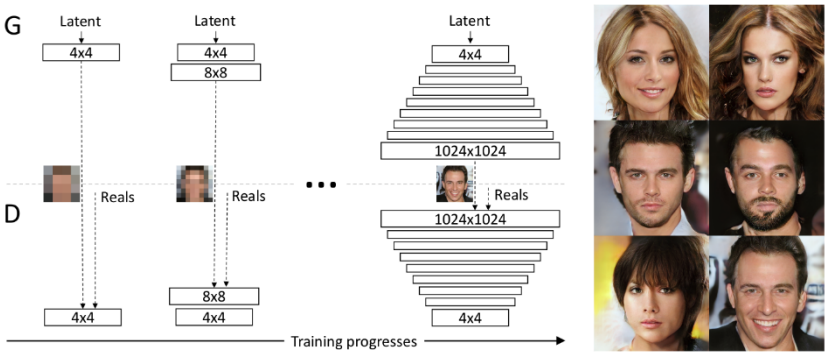


Figure 7: ProgressiveGAN architecture schematic to generate HD photorealistic images of human faces: The layers of the network are progressively increased while the training progresses (left). (Karras, Aila, Laine, & Lehtinen, 2017)

Progressive GAN is the baseline architecture for the famous styleGAN (Karras, Laine, & Aila, 2019), which uses an extra mapping network induced into the generator with the so-called AdaIN operations to better disentangle the latent factors of variation. This improvement enables the intuitive and scale-specific control of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation (e.g., freckles and hair), thereby reaching outstanding results in generating HD photorealistic images of human faces and style transfer. Further improvements of the styleGANs can be seen in StyleGAN2 (Karras, et al., 2020) and StyleGAN2-ADA (Karras, et al., 2020).

As we are dealing with translation of a continuous stream of sonar and optical images in our case, the generated sequences of images not only must be photorealistic individually but also temporally consistent as a whole. Vid2vid [ređ] is an extension of pix2pix to cope with time consistency for a sequence of images, where the authors impose the condition that the generated image not only depends on the current source image s_t but also on the past source sequence of images s^(t-1)_t-T as well as generated sequence of images x^(t-1)_t-T. Additionally, the authors make use of the inherent redundant information of consecutive images, by combining the resulting synthesized intermediate image h_t with an optical-flow warped version of the last generated image w_t by means of a soft occlusion map for attention-based aggregation m_t.

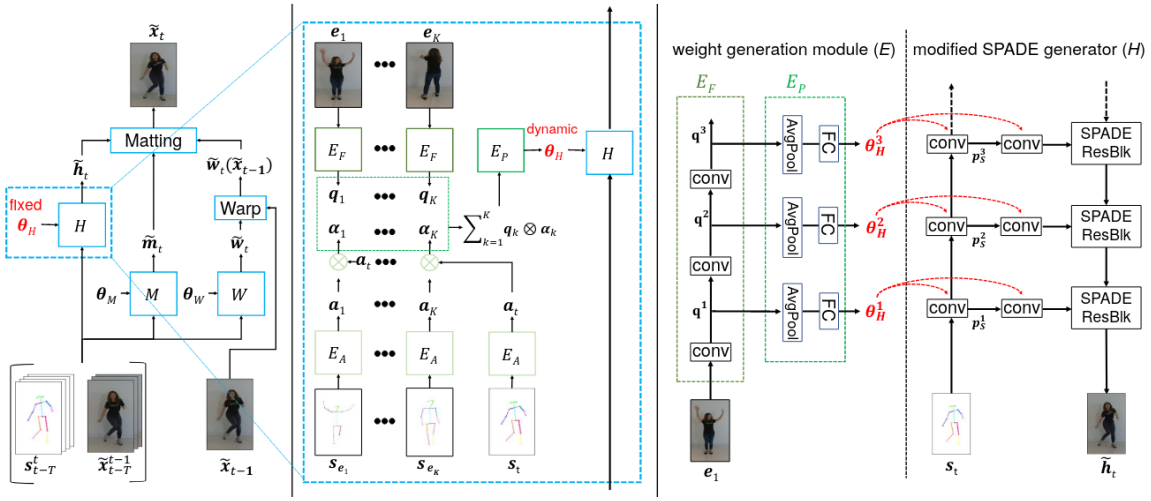


Figure 8: vid2vid framework schematic for a few-shot training (Wang, et al., 2018)

H2020-ICT-47-2020 Grant Agreement Number 101016958

All the above-mentioned techniques applied to improve the original pix2pix architecture will be deeper investigated during the development of the SONAVision algorithm. The possibility of applying widely used NN to the generated realistic visual-like (depth) image for relevant computer vision applications, such as object/human detection [Lie et al., 2020] and human pose estimation (Dang, Yin, Wang, & Zheng, 2019), will be investigated as well. The applicability of those methods will be used as a criterion, among others, to evaluate the realism in the reconstruction of the generated image.

Detection and pose estimation with sonar images

Object detection in sonar images is a well studied field, and in this project, we will need basic object detection capabilities to detect human divers (UC1) and obstacles (UC2). For this purpose, we will use methods from the state of the art, including detection proposals (for obstacle detection in UC2), and deep convolutional object detectors like SSD, Faster R-CNN, and variations of YOLO.

This work also connects to self-supervised learning and the development of pre-trained models, as they are required for object detection models to train and perform correctly. For safe use and active learning, we would also produce ensembles of object detectors for uncertainty quantification, tuning the architecture and method to maximize computational performance while keeping good epistemic uncertainty quantification.

Self-supervised Learning:

Due to the lack of available labelled sonar data, we will investigate the application of self-supervised learning techniques to increase performance in downstream tasks of interest, such as translation, diver detection and pose estimation. This applies to all use cases that use machine learning and neural network models, as we believe, they can be pre-trained using self-supervised learning to learn basic concepts in a sonar image, and hence, improve performance in the final downstream task. Although self-supervised learning has been demonstrated to improve performance in colour images for many tasks (Jing & Tian, 2020), it has not yet been applied to sonar images to the best of our knowledge.

The main principle of designing pretext tasks is to find a suitable task which is not too difficult and not too easy for a network to solve. If it is too difficult, the network may not converge due to the ambiguity of the task, and if it too easy, the network will learn trivial solutions, leading to a barely improvement in the downstream task. Furthermore, the pretext tasks must ensure in our case that spatial and temporal features are learned through the process of accomplishing the pretext task, as these are key for the aimed downstream tasks of translation, diver detection and pose estimation.

In the project, we will explore several pretext tasks, starting with the ones already widely used for colour images that we believe can be applied for sonar images with similar performance. If required, we will develop new possible pretext tasks for sonar images exclusively. All the pretext tasks will be evaluated and benchmarked with the three aimed downstream tasks and classification.

The existing pretext tasks for colour images that we will mainly explore are (i) generation-based methods, such as super-resolution [Ledig et al., 2017], in-painting [Pathak et al., 2016] and video prediction (Srivastava, Mansimov, & Salakhudinov, 2015) (in our case prediction of the following sonar image in a frame sequence), (ii) spatial context methods, such as solving the jigsaw puzzle (Noroozi, Favaro, & Springer, 2016) (Taleb, Lippert, Klein, & Nabi, 2021) and recognizing rotations (Komodakis & Gidaris, 2018), and (iii) temporal context methods,

H2020-ICT-47-2020 Grant Agreement Number 101016958

such as recognizing the order of the frame sequence (Misra, Zitnick, & Hebert, 2016) (Lee, Huang, Singh, & Yang, 2017).

Uncertainty:

In this project we plan to use uncertainty in ML as a way to enable the THW operator to interpret the results and obtain reliable confidence estimates. This usage has the advantage of adding a layer of safety in cases of false positive or negative detections of divers, which are undesirable.

Uncertainty quantification can also be used for active learning, where the model can inform the operator if there are samples that could benefit the model (with high uncertainty) and could be labelled and incorporated in future training instances. We plan to use active learning in UC1.

A final use of uncertainty in ML applied to UC2, where an ensemble of neural networks can be used to detect novel or "unknown" objects that are potentially obstacles which are not covered by the training set, indicating the AUV and the operator of potential cases that should be considered in future versions of the models.

3.2 EagleEye

3.1.2 Goals

EagleEye aims to leverage information from the scenario of co-located camera and a forward-looking sonar. This is a typical scenario as Forward-Looking Sonars (FLS) and Forward-Looking Cameras (FLC) are used in conjunction in many under water platforms because of their complementary abilities. The FLS has a long range, and its performance does not depend on water conditions. However, it produces low resolution images that lack vertical information, and its input is unstable in short ranges due to reverberations. On the other hand, the FLS has excellent spatial resolution with colour information but only works in short ranges.

Because of the FLS mode of operation, this yields two very different viewpoints (illustrated in Figure X). The FLS displays a top view 2D acoustic image of the scene (Figure X right), whereas the FLC provides a 2D side-view optical image (Figure X middle), despite being located in proximity (Figure X left).

The task of combining these two very different modalities in such unaligned point of views is very challenging and so far, there have been little relevant work on the topic. Because of the viewpoint change here our goal is not to do a direct registration or translation of the images, but to co-locate objects from the two different viewpoints.

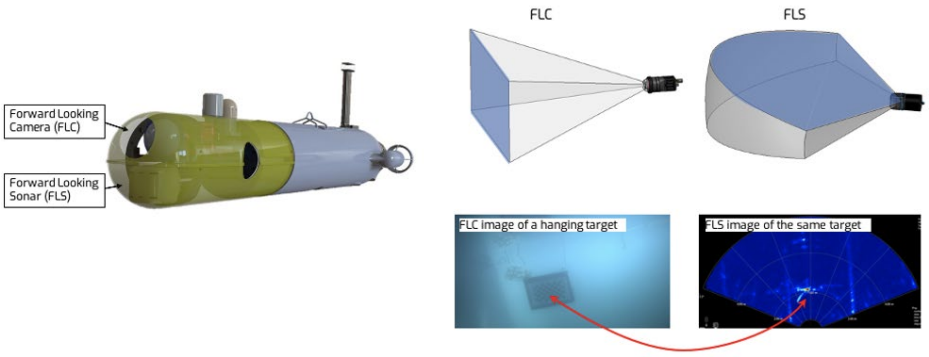


Figure 9: EagleEye concept

H2020-ICT-47-2020 Grant Agreement Number 101016958

Relation to the multi-modal learning challenges:

- The main challenge we face is representation: how can both modalities be represented such that the same object can be identified in both.
- As a step towards solving representation we plan to perform an implicit alignment of viewpoints.

3.2.2 Methods:

We draw inspiration from recent publications that describe the use of deep learning to match aerial images with street-view images (Hu, Feng, Nguyen, & Lee, 2018) (Lin, Cui, Belongie, & Hays, 2015) (Regmi & Shah, 2019) (Tian, Chen, & Shah, 2017) (Figure 10 left). Another relevant topic is matching architecture floor plans to actual images from the apartment (Liu, Wu, Kohli, & Furukawa, 2016) (Figure 10 right).

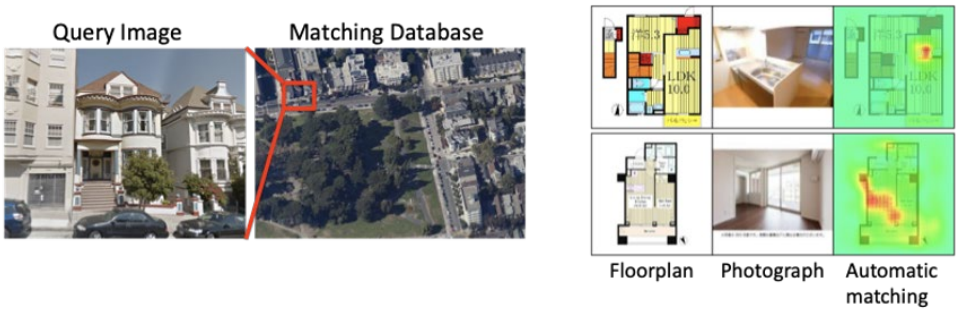


Figure 10: EagleEye analogues in terrestrial applications

These two applications, and especially the latter one, resemble our case as they match images from completely different viewpoints: front looking and a top view. The floorplan matching problem even uses images from two different modalities. However, our case is even more difficult because the FLS image has a much lower resolution and a high noise level. We plan to compensate for that by adding physical information on the calibration between the two sensors, such as the calibration geometry and intrinsic parameters of each sensor, into the network.

GANs:

(Regmi & Shah, 2019) suggest that training a GAN network to generate the second view can significantly improve the detection of objects by feeding the original view and the generated one to the detection pipeline. They do that by adopting the X-Fork generator architecture (Regmi & Borji, 2018) to train the GAN for cross-view image synthesis. The X-Fork is a multi-task learning architecture that synthesizes cross-view image as well as semantic segmentation map. Similarly, we also plan to develop a GAN network to transform optical images to top-view images.

H2020-ICT-47-2020 Grant Agreement Number 101016958

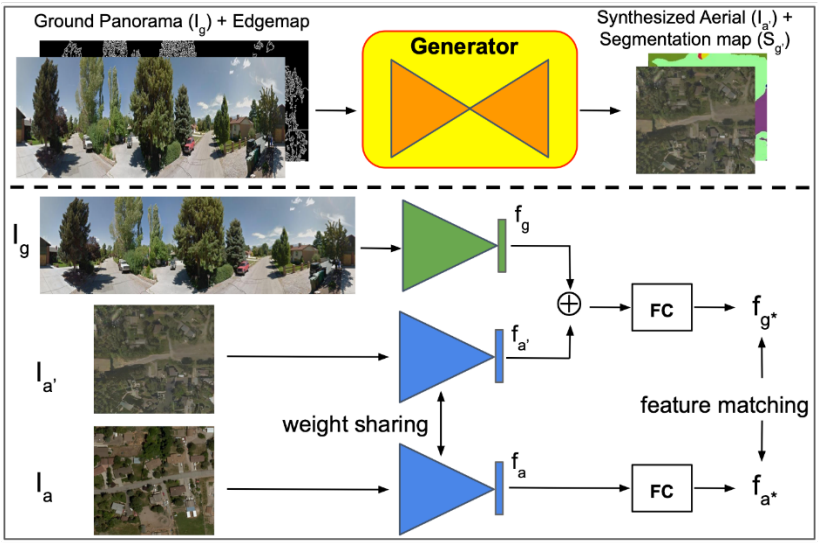


Figure 11: (Regmi & Shah, 2019) schematic algorithm.

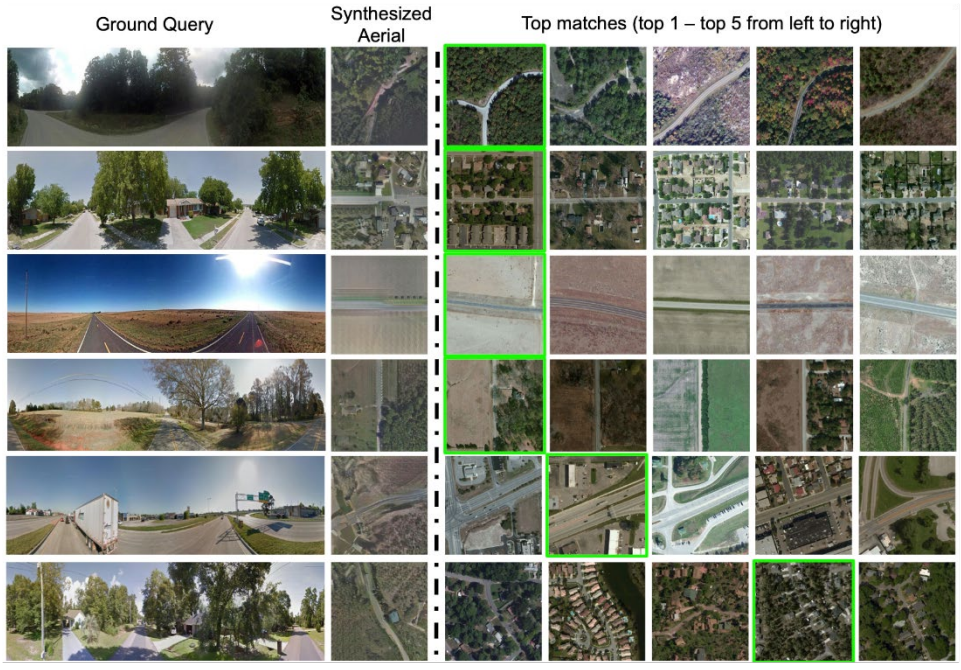


Figure 12: (Regmi & Shah, 2019) Example results.

CNNs:

A CNN will be trained to take three inputs: an optical image, its simulated top view image from the GAN, the FLS image, and output the similarity between the optical and the acoustic image (see Fig. X bottom).

As baseline for the CNN architecture, we will use the one presented by (Liu, Wu, Kohli, & Furukawa, 2016). This network extracts the similarities between the inputs, i.e., an image and a floor plan, in order to find the location of the object in the floor plan. In order to highlight the object in the floor plan, the authors utilize the receptive

H2020-ICT-47-2020 Grant Agreement Number 101016958

field of the network using the same method of (Zhou et al. 2015) (Fig. 13). A high value of the receptive field indicates the location of the photographed object in the floor plan.

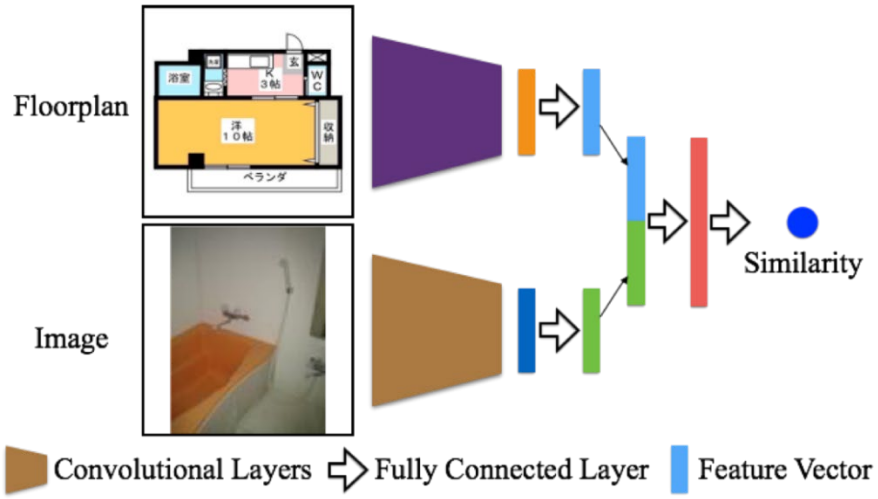


Figure 13: Network structure of (Liu, Wu, Kohli, & Furukawa, 2016).

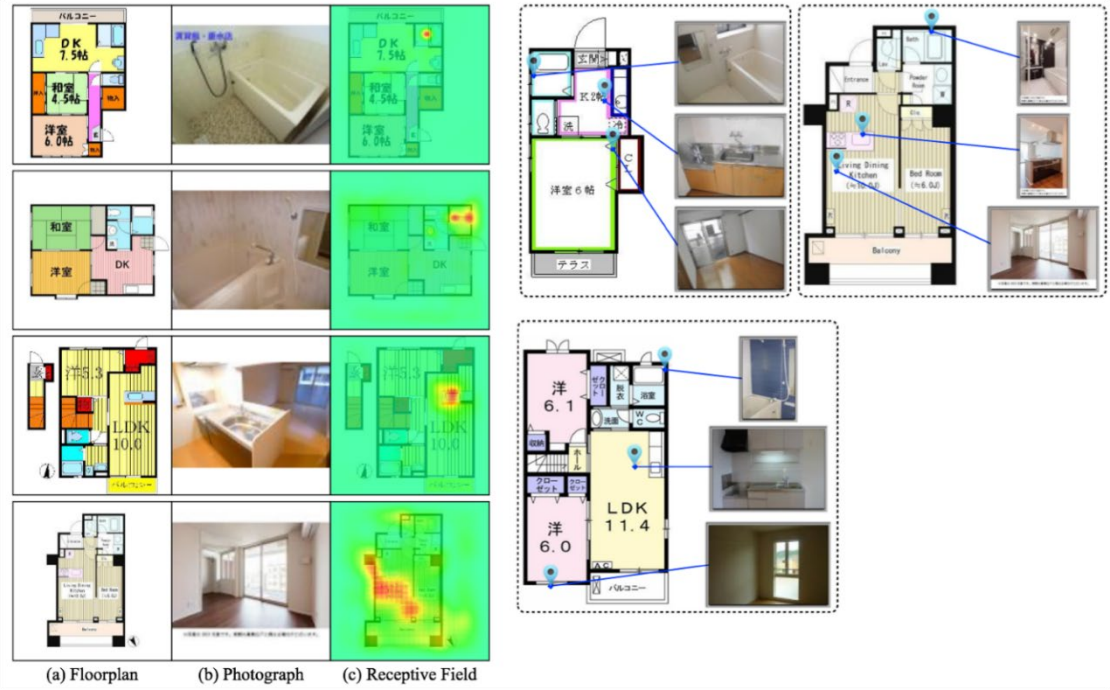


Figure 14: Visualizing receptive fields for object detection in floor plans.

Transformers:

As for other tasks in deep learning transformers are starting to be used for the task of image registration, e.g., (Wang & Delingette, 2021). We will consider using this framework for the 2nd stage of the detection, to replace the CNN.

3.3 SmartSeafloorScan

H2020-ICT-47-2020 Grant Agreement Number 101016958

Goals:

The central objective of the SmartSeafloorScan algorithms is to perform automatic seabed classification by exploiting acoustic and optical imagery. Optical imagery will be used for training, classification and benchmarking with the overall objective of allowing, by the end of the project, to perform accurate sea-floor characterization without the need for optical imagery.

Distinguishing marine benthic habitat characteristics is of key importance in the planning and deployment of seafloor installations (from oil and gas installations to planning pipelines or cabling for the energy industry). While sidescan and multibeam sonar allow for distinguishing between hard and soft seafloor, the classification of finer sediments (such as mud/sand/clay) requires nowadays the deployment of sampling cruises in which core samples are brought to the surface for examination. Therefore, the standard process of seafloor classification currently requires with a Geophysical survey, followed by a sampling survey.

The SmartSeafloorScan algorithm will use explore the techniques describe in the following paragraphs to perform multisensor fusion. Instead of relying on cores, optical data will be used to ground truth the acoustic reflectivity perceived by the side-scan sonar, and it will be combined with navigation data and multibeam sonar (if available), to provide accurate range estimates for every point in the sidescan sonar image.

Relation to the multi-modal learning challenges:

An obvious challenge for our use case is the fusion of the two modalities to generate predictions. The approach we propose is to train two distinct unimodal classifiers either using DNNs or a classical ML approach such as SVMs and employ a fusion mechanism such as majority voting or weighted averaging of the predictions made by these classifiers to generate the final output. Then, using this as our baseline we adopt different strategies to further improve the quality of predictions. A proposition would be to employ co-learning between side-scan sonar and camera-based inputs using modality dropouts, which has been proven to result in better predictions in the domains of audio-visual information processing (Hussen Abdelaziz, et al., 2020) and medical image processing (Li, et al., 2018). These approaches, however, raise yet another challenge of aligning the side-scan sonar and camera viewpoints. Since the training data was not collected in parallel i.e., the samples of both modalities were collected as part of different survey missions, albeit spanning the same area, the idea is to use the available navigation information to align the image patches drawn from the two modalities.

Methods:

GANs:

One approach to deal with class imbalance and the lack of data in general is to exploit well regarded architectures such as cycle-GANs [Huang et al., 2018], pix2pix (Isola, Zhu, Zhou, & Efros, 2017) or style-GANs (Karras, Laine, & Aila, 2019) to synthetically generate the additional data for training. This can either be used as a data augmentation strategy or directly be used to expand the dataset. Further, with regards to SSS imagery, the brightness, contrast and other attributes vary quite significantly across different types of sonar instruments used to collect data and the various strategies adopted to process the raw data, which might add some bias to the classifier. With the use of GANs, however, one can generate synthetic data that accommodates all these different representations, resulting in a much more generalized and compatible classifier as suggested by [*Li et al., 2019*].

Transformers:

Considering the inherent limitations of CNNs as previously discussed and the amount of attention that Transformers have been garnering in overcoming those limitations, a viable approach would be to disregard the use of CNNs altogether. Instead of relying on Unet, the de facto standard in semantic segmentation, we want to investigate TransUnet (Chen, et al., 2021), a hybrid CNN-Transformer U-shaped Encoder-Decoder architecture that not only benefits from the global context of transformers, leveraging self-attention in computer vision, but also from precise localization of high resolution CNN feature maps incorporated using skip connections; and SwinUnet (Cao, et al., 2021) which follows a more straightforward approach of using a pure transformer-based U-shaped Encoder-Decoder architecture which has shown superior performance in medical image segmentation.

Self-supervised Learning:

Another approach to deal with the lack of sufficient labelled data on the actual task at hand would be to resort to unsupervised representation learning and then fine-tuning the learned representations using the available data on the actual task. SimSiam (Chen, Kornblith, Norouzi, & Hinton, 2020) and SwAV (Caron, et al., 2020) are two such approaches that we plan to investigate, owing to their superior performance and computational efficiency, to train our unimodal networks. Furthermore, since the use of Transformers in self-supervised learning models has shown to boost the quality of predictions even more (Caron, et al., 2021) (Chen, et al., 2021), we plan to incorporate them on the simpler and more performant SSL variants namely SimSiam and SwAV to try and get a further increase in performance.

4 Framework Concept

4.1 Design Rationale

The main goal behind the development of a DeeperSense framework is to provide a self-contained platform for the development, evaluation and deployment of multi-modal deep learning technologies in the context of robotic perception in marine environments. This framework is meant to encompass the three algorithms described in Section 3 in a unified yet modular software library, that allows the development of a processing pipeline[1] by an end user, as well as the deployment of a fully trained[2] pipeline onto an underwater robotic platform.

The process of designing the framework architecture will take into consideration the requirements of the usecases that have been described in the deliverable D2.2. Additionally, the concepts presented in this document has been discussed to achieve a common consensus by the members of the consortium that are

[1] By the term *pipeline*, we refer to a processing chain that includes one of algorithms described above, together with pre- and post-processing tools.

[2] We differentiate between untrained and trained models. An *untrained model* is the default randomly initialized network that has not been trained or optimized with data, and a *trained model* is a network that has been trained with data using a certain optimization method.

involved in the software development. As a result of these discussion, a concept of a framework architecture has been devised that is presented in the following paragraphs.

4.2 Architecture Concept

The architecture concept of the DeeperSense framework will follow two main design goals: (1) to be easily approachable and quickly deployable, while (2) being highly tweakable. To achieve these features, the framework will follow a layered architecture, from a low-level module providing the core building blocks of the described algorithms, followed by higher-level modules providing the tools and functionality to customize, train and deploy the desired algorithms. Additionally, the framework will provide an extra optional module that provides products to visualize and compare the results of different models.

A schematic representation of the framework is shown in Figure 16. The framework will be composed of three main packages Deepersense-core, DeeperSense-dev, DeeperSense-ros, and an optional package Deepersense-utils. At the lowest level of the framework is the DeeperSense-core package which provides the core deep learning and data processing methods for each of the three algorithms described above. DeeperSense-dev will represent the higher-level API that provides the user with the necessary tools to construct the processing pipelines relevant to any of the usecases, load the desired dataset, and train and validate the constructed processing pipeline. DeeperSense-ros is also considered as a higher-level module. It will represent the main interface to the robotic platform, where a model that has been trained by a user could be deployed on the platform as a dedicated ROS node. Finally, DeeperSense-utils will represent an optional package that could be used by the user for comparing the performance of different models and for visualization purposes.

A more detailed description of each of package is provided next.

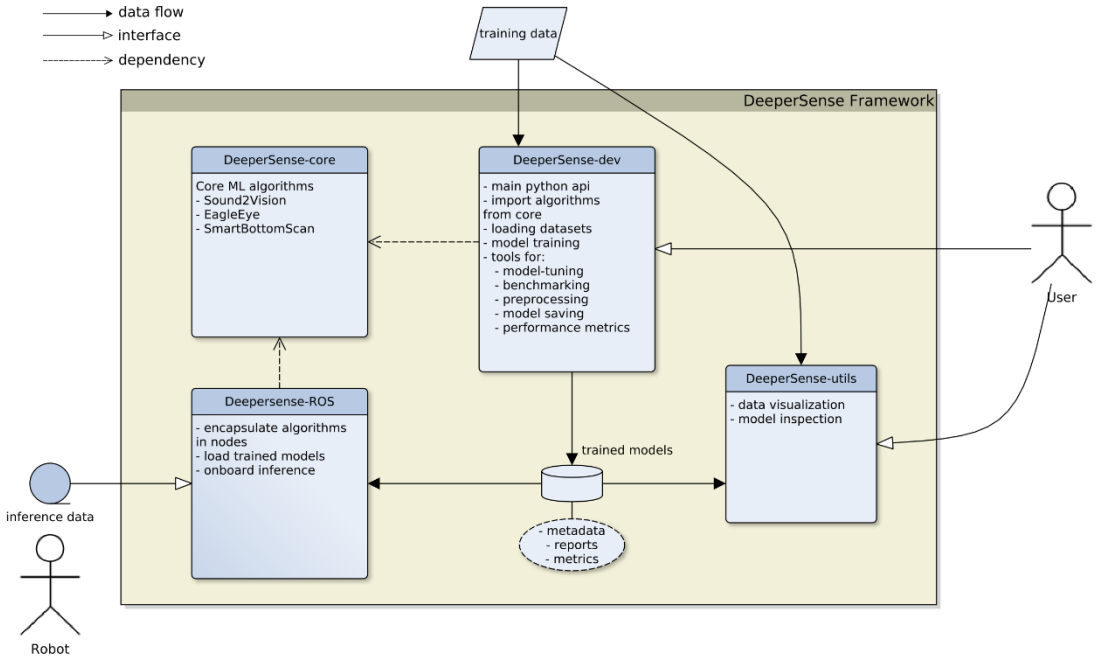


Figure 15: DeeperSense framework concept.

DeeperSense-core

DeeperSense-core provides the "core" set of state-of-the-art algorithms and techniques described in section 3, as a ready-to-use library or package. These algorithms represent the deep learning methods necessary to fuse sensory data. DeeperSense-core will be implemented in a modular fashion, where a common interface will be developed to allow the configuration and deployment of any of the algorithms as a distributed system on multiple platforms. The algorithms provided by this package are to be deployed on a designer's environment as well as on a target robotic system.

This module will be built on top of optimized lower-level libraries such as Tensorflow **(Abadi, et al., 2016)**, Keras **(Chollet, 2007)**, PyTorch **(Paszke, et al., 2017)**, Scikit-learn **(Pedregosa, et al., 2011)**, NumPy **(Oliphant, 2006)**, pandas **(McKinney, others, & Austin, TX, 2010)** and others.

Each of the DeeperSense algorithms will be implemented as its own class that defines the basic architecture of the algorithm. Any variations of a certain algorithms can be implemented as an inheritance from the base algorithm. To allow for customization of a certain algorithm to fit a certain application of dataset, several attributes will be exposed to the user as properties that could be set on instantiating the algorithm. Such properties could be attributes such as model layers and node types, optimizers, callbacks, loss functions, etc.

The API of DeeperSense-core is aimed to be as user-friendly as possible will full documentation and basic know-how examples. It will be made available for users to build their custom pipelines on top or modify the ones provided by default.

DeeperSense-dev

DeeperSense-dev is a high-level API that will provide the main developer tools for the user to build, train and test algorithms relevant to each of the usecases tackled in this project. It will represent the main interface to the user to (1) import and instantiate core algorithms from provided in *DeeperSense-core*, (2) load datasets from the data repository that will be generated in WP4, and (3) train and validate the corresponding algorithms.

Model prototyping

Any core algorithm provided in *DeeperSense-core* could be imported as a component, where a user can instantiate with a desired configuration. As each algorithm may have a fixed topology, the user could still configure hyper-parameters such as the number of layers, number of nodes in each layer, activation functions and the kind of optimizer. After an algorithm is instantiated and built successfully, it can be then ready to be trained.

Data loading and Pre-processing

In most machine learning practices, loading and pre-processing data is yet an ad-hoc process. To facilitate the model training, a set of standard operations will be provided that could be defined as:

Loading data into working memory: loading the data from the source is the most basic operation. For this purpose, methods for loading human readable data, as well as de-serialization methods for the case of machine-readable binary data.

Pre-processing (normalization): pre-processing and filtering tools necessary to transform the data into a format compatible with the input of the desired algorithm.

Data splitting: splitting the data into training, validation, and testing sets, and cross-validation schemes.

This package will also provide the means to save processed data to be used for other applications.

Model tuning

Tools for tuning the hyper-parameters of the desired algorithms for optimal performance will be provided by this package. This requires methods for searching or sampling from the candidate hyper-parameter space, a cross-validation scheme, and metrics for evaluating the performance of a candidate estimator.

Model saving/loading

By training a certain algorithm, a model or a hypothesis is created that is described by (1) the topology of the selected algorithm, (2) the model parameters or weights resulting from the training, and a set of hyper-parameters that were used during this process. *DeeperSense-dev* will provide the means for saving (serializing) a trained model along with its corresponding meta-data as a file (for example HDF5) that can be loaded for future use without having to retrain.

Products

Trained models: the main product of DeeperSense-dev are the models resulting from training using the data collected in WP4. A trained model will be exported as a serialized portable data file representing the architecture of the used algorithm and its associated weights or parameters.

Model metadata: along with each trained model, a file containing the meta-information will be provided. This file will contain the necessary information to identify the model, for example which dataset was used, how the model has been trained (training/validation split, optimizer used, callbacks, number of epochs, etc.), and other hyperparameters used.

Reports & metrics: additional information reporting on the quality of the trained model will be exported with each trained model. This will include training logs consisting of the training and validation loss per epoch, as well as evaluation scores and metrics resulting from evaluating the trained model on test data.

DeeperSense-ROS (interface to robot)

DeeperSense-ROS will provide ROS nodes that encapsulate the core algorithms provided in DeeperSense-core. This package represents the main interface to a robotic platform, where provided nodes are to be deployed on the robot with the goal of performing on-board predictions.

A node will load a saved model that was trained by a user to perform a task relevant to a certain usecase addressed in this project. In general, the nodes provided in DeeperSense-ROS are meant to be only used to perform predictions using a pretrained model. In some case however, on-board training could be made possible provided that the node designer chooses to implement on-line training features, and that the robot is equipped with the necessary computational hardware and energy resources.

Model compression will be implemented in this package with the aim at achieving real-time performance of the trained network on deployment, however in some cases this might not be guaranteed due to the experimental nature of the developed algorithms.

This package will also provide the capability of logging the output generated by the deployed algorithms for offline post-processing and benchmarking. Additionally, an adapter that connects the output of the ROS node will be provided for visualization and inspection purposes.

DeeperSense-utils:

DeeperSense-utils will be a high-level API that will provide utility tools to inspect trained models as well as plotting and visualizations. This package will be designed as an optional package that a user can choose to install or not, without affecting the main functionality of the framework.

To inspect the validity of trained models, this package will include methods for evaluating the performance as well as the uncertainty and interpretability of the trained models. In any supervised training application, a model is prone to overfitting if it was poorly tuned, or not enough data is provided. Over-fitting is the situation when a model can perfectly repeat the data sample that it has seen during training but fails to predict something useful on new samples.

Model uncertainty is another issue that is often overlooked when training machine learning algorithms. Uncertainty could be aleatoric, i.e., inherent to the data, or epistemic which is attributed to a badly designed model or insufficient training data.

DeeperSense-utils will also provide plotting tools producing graphical representations for model inspection and comparison. This would include tools such as validation and training curves. Validation curves can be useful to inspect the sensitivity of a model when varying a certain hyperparameter. Training curves are used to assess the sample efficiency of a model, by plotting the training and validation curves against the number of training samples. If by increasing the number of training data both scores converge to a similar value, this would be an indication that adding more training samples will not add any further benefit to the model.

5 Conclusions

In this document an extensive review of the state-of-the-art of multi-modal deep learning literature was provided, describing five main challenges of this field. Based on this review, a number of methods were identified in this document to address the requirements of the different usescase tackled in Deepersense. The goals of each algorithm were stated and their relation to the five challenges of multi-modal learning were discussed.

For the SONAVision algorithm, the concept of an end-to-end association method between sonar and visual camera images observing the same underwater scene was detailed. Several variants of Generative Adversarial Networks (GANs) were proposed to be evaluated for the task of translation between sonar and camera images. Self-supervised techniques will be adopted in SONAVision, in order to mitigate the issue of lack of training data and improve the performance of the algorithm.

In the case of EagleEye, the aim is to co-locate objects in sonar and visual images resulting from the two different viewpoints. For this purpose, methods such as Convolutional Neural Networks (CNNs), GANs and Transformers will be evaluated.

The SmartSeafloorScan algorithms aims at performing automatic classification of seabed terrain by exploiting acoustic and optical imagery. In addition to GAN architectures, attention-based networks such as transformers are selected to be evaluated for this task. For this usecase, self-supervised techniques will also be used to increase the classification performance due to the lack of labelled data.

Finally, a concept of a common framework to encapsulate the implementations of the above-mentioned algorithms was proposed. The aim of this framework is to provide the user with the tools for developing and training of the processing pipelines for each task, as well as the deployment of the trained models on a robotic platform using the Robot Operating System (ROS) as a middleware.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . others. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Anwar, S., Hwang, K., & Sung, W. (2017). Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC), 13*(3), 1-18.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. *International conference on machine learning*, (pp. 214-223).

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems, 29*, 892-900.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A., & Springer. (2011). Sequential deep learning for human action recognition. *International workshop on human behavior understanding*, (pp. 29-39).

Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence, 41*(2), 423-443.

Blalock, D., Ortiz, J. J., Frankle, J., & Guttag, J. (2020). What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*.

Bregler, C., Covell, M., & Slaney, M. (1997). Video rewrite: Driving visual speech with audio. *ConferenceProceedings of the 24th annual conference on Computer graphics and interactive techniques*, (pp. 353-360).

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Burgard, W., Valada, A., Radwan, N., Naseer, T., Zhang, J., Vertens, J., . . . Oliveira, G. (2020). Perspectives on deep multimodel robot learning. Springer.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2021). Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. *arXiv preprint arXiv:2105.05537*.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*.

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., . . . Zhou, Y. (2021). Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International conference on machine learning*, (pp. 1597-1607).

Chen, X., & He, K. (2021). Exploring simple siamese representation learning. *ConferenceProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 15750-15758).

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *ConferenceProceedings of the 30th International Conference on Neural Information Processing Systems*, (pp. 2180-2188).

Chen, X., Xie, S., & He, K. (2021). An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*.

Chollet, F. (2007). others.(2015). Keras. GitHub.

Christensen, J. H., Hornauer, S., Stella, X. Y., & IEEE. (2020). BatVision: Learning to see 3D spatial layout with two ears. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 1581-1587).

Dang, Q., Yin, J., Wang, B., & Zheng, W. (2019). Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology, 24*(6), 663-676.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., . . . Mitchell, M. (2015). Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*.

Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . others. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gao, R., Chen, C., Al-Halah, Z., Schissler, C., Grauman, K., & Springer. (2020). Visualechoes: Spatial image representation learning through echolocation. *European Conference on Computer Vision*, (pp. 658-676).

Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems, 27*.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., . . . others. (2020). Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Guo, Y., Zhang, C., Zhang, C., & Chen, Y. (2018). Sparse dnns with improved adversarial robustness. *arXiv preprint arXiv:1810.09619*.

Gupta, A., Verma, Y., & Jawahar, C. (2012). Choosing linguistics over vision to describe images. *ConferenceProceedings of the AAAI Conference on Artificial Intelligence*, *26.*

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. *ConferenceProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 9729-9738).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hou, L., Yao, Q., & Kwok, J. T. (2016). Loss-aware binarization of deep networks. *arXiv preprint arXiv:1611.01600*.

Hu, S., Feng, M., Nguyen, R. M., & Lee, G. H. (2018). Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. *ConferenceProceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 7258-7267).

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *ConferenceProceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4700-4708).

Hunt, A. J., Black, A. W., & IEEE. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference ConferenceProceedings*, *1*, pp. 373-376.

Hussen Abdelaziz, A., Theobald, B.-J., Dixon, P., Knothe, R., Apostoloff, N., & Kajareker, S. (2020). Modality dropout for improved performance-driven talking faces. *ConferenceProceedings of the 2020 International Conference on Multimodal Interaction*, (pp. 378-386).

Hussen Abdelaziz, A., Theobald, B.-J., Dixon, P., Knothe, R., Apostoloff, N., & Kajareker, S. (2020). Modality dropout for improved performance-driven talking faces. *Proceedings of the 2020 International Conference on Multimodal Interaction*, (pp. 378--386).

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *ConferenceProceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 1125-1134).

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., . . . Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *ConferenceProceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 2704-2713).

Jing, L., & Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

Karnewar, A., & Wang, O. (2020). Msg-gan: Multi-scale gradients for generative adversarial networks. *ConferenceProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 7799-7808).

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. *ConferenceProceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 3128-3137).

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *ConferenceProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 4401-4410).

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. *ConferenceProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 8110-8119).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Komodakis, N., & Gidaris, S. (2018). Unsupervised representation learning by predicting image rotations. *International Conference on Learning Representations (ICLR).*

Kuznetsova, P., Ordonez, V., Berg, A., Berg, T., & Choi, Y. (2012). Collective generation of natural image descriptions. *ConferenceProceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (pp. 359-368).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature, 521*(7553), 436-444.

Lee, H.-Y., Huang, J.-B., Singh, M., & Yang, M.-H. (2017). Unsupervised representation learning by sorting sequences. *ConferenceProceedings of the IEEE International Conference on Computer Vision*, (pp. 667-676).

Li, C., Wand, M., & Springer. (2016). Precomputed real-time texture synthesis with markovian generative adversarial networks. *European conference on computer vision*, (pp. 702-716).

Li, C., Wand, M., & Springer. (2016). Precomputed real-time texture synthesis with markovian generative adversarial networks. *European conference on computer vision*, (pp. 702-716).

Li, K., Yu, F., Wang, Q., Wu, M., Li, G., Yan, T., . . . IEEE. (2019). Real-Time Segmentation of Side Scan Sonar Imagery for AUVs. *2019 IEEE Underwater Technology (UT)*, (pp. 1-5).

Li, R., Wang, C., Liu, S., Wang, J., Liu, G., & Zeng, B. (2021). UPHDR-GAN: Generative Adversarial Network for High Dynamic Range Imaging with Unpaired Data. *arXiv preprint arXiv:2102.01850*.

Li, X., Dou, Q., Chen, H., Fu, C.-W., Qi, X., Belav`y, D. L., . . . Heng, P.-A. (2018). 3D multi-scale FCN with random modality voxel dropout learning for intervertebral disc localization and segmentation from multi-modality MR images. *Medical image analysis, 45*, 41-54.

Lin, T.-Y., Cui, Y., Belongie, S., & Hays, J. (2015). Learning deep representations for ground-to-aerial geolocalization. *ConferenceProceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 5007-5015).

Liu, C., Wu, J., Kohli, P., & Furukawa, Y. (2016). Deep multi-modal image correspondence learning. *arXiv preprint arXiv:1612.01225*.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision, 128*(2), 261-318.

Liu, M.-Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, (pp. 700-708).

Mansimov, E., Parisotto, E., Ba, J. L., & Salakhutdinov, R. (2015). Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*.

Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. *ConferenceProceedings of the IEEE international conference on computer vision*, (pp. 2794-2802).

McKinney, W., others, & Austin, TX. (2010). Data structures for statistical computing in python. *ConferenceProceedings of the 9th Python in Science Conference*, *445*, pp. 51-56.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., & Ghasemzadeh, H. (2020). Improved knowledge distillation via teacher assistant. *ConferenceProceedings of the AAAI Conference on Artificial Intelligence*, *34*, pp. 5191-5198.

Mishra, R., Gupta, H. P., & Dutta, T. (2020). A survey on deep neural network compression: Challenges, overview, and solutions. *arXiv preprint arXiv:2010.03954*.

Misra, I., Zitnick, C. L., & Hebert, M. (2016). Shuffle and learn: unsupervised learning using temporal order verification. *European Conference on Computer Vision* (pp. 527-544). Springer.

Noroozi, M., Favaro, P., & Springer. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. *European conference on computer vision*, (pp. 69-84).

Odena, A., Olah, C., Shlens, J., & PMLR. (2017). Conditional image synthesis with auxiliary classifier gans. *International conference on machine learning*, (pp. 2642-2651).

Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.

Ordonez, V., Kulkarni, G., & Berg, T. (2011). Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems, 24*, 1143-1151.

Ouali, Y., Hudelot, C., & Tami, M. (2020). An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*.

Park, T., Efros, A. A., Zhang, R., Zhu, J.-Y., & Springer. (2020). Contrastive learning for unpaired image-to-image translation. *European Conference on Computer Vision*, (pp. 319-345).

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. *ConferenceProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 2337-2346).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., . . . Lerer, A. (2017). Automatic differentiation in pytorch.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. *ConferenceProceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 2536-2544).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . others. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H., & PMLR. (2016). Generative adversarial text to image synthesis. *International Conference on Machine Learning*, (pp. 1060-1069).

Regmi, K., & Borji, A. (2018). Cross-view image synthesis using conditional gans. *ConferenceProceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3501-3510).

Regmi, K., & Shah, M. (2019). Bridging the domain gap for ground-to-aerial image matching. *ConferenceProceedings of the IEEE/CVF International Conference on Computer Vision*, (pp. 470-479).

Rohrbach, A., Rohrbach, M., Schiele, B., & Springer. (2015). The long-short story of movie description. *German conference on pattern recognition*, (pp. 209-221).

Sak, H., Senior, A., Rao, K., & Beaufays, F. (2015). Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*.

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

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. *International conference on machine learning*, (pp. 843-852).

Taleb, A., Lippert, C., Klein, T., & Nabi, M. (2021). Multimodal self-supervised learning for medical image analysis. *International Conference on Information Processing in Medical Imaging* (pp. 661-673). Springer.

Terayama, K., Shin, K., Mizuno, K., & Tsuda, K. (2019). Integration of sonar and optical camera images using deep neural network for fish monitoring. *Aquacultural Engineering, 86*, 102000.

Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., & Mooney, R. (2014). Integrating language and vision to generate natural language descriptions of videos in the wild. *ConferenceProceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, (pp. 1218-1227).

Tian, Y., Chen, C., & Shah, M. (2017). Cross-view image matching for geo-localization in urban environments. *ConferenceProceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3608-3616).

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2020). Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, (pp. 5998-6008).

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., & Saenko, K. (2014). Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., & Catanzaro, B. (2018). Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. *ConferenceProceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 8798-8807).

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., . . . Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. *ConferenceProceedings of the European conference on computer vision (ECCV) workshops*, (pp. 0-0).

Wang, Z., & Delingette, H. (2021). Attention for Image Registration (AiR): an unsupervised Transformer approach. *arXiv preprint arXiv:2105.02282*.

Xingjian, S. H., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, (pp. 802-810).

Xu, R., Xiong, C., Chen, W., & Corso, J. (2015). Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. *ConferenceProceedings of the AAAI Conference on Artificial Intelligence*, *29*.

Zhang, R., Isola, P., Efros, A. A., & Springer. (2016). Colorful image colorization. *European conference on computer vision*, (pp. 649-666).

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2014). Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *ConferenceProceedings of the IEEE international conference on computer vision*, (pp. 2223-2232).

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *ConferenceProceedings of the IEEE international conference on computer vision*, (pp. 2223-2232).

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E. (2017). Multimodal Image-to-Image Translation by Enforcing Bi-Cycle Consistency. *Advances in neural information processing systems*, (pp. 465-476).

DeeperSense Consortium