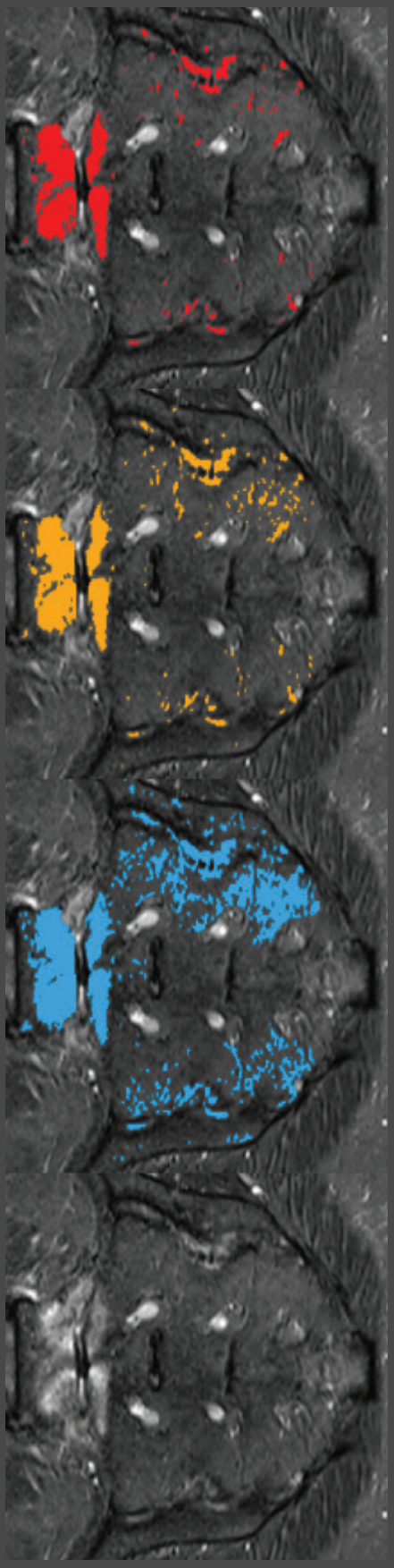
July 2021

Computer Vision News

The Magazine of the Algorithm Community



Our Exclusive Review of the Latest Great Paper by Piotr Dollár and FAIR

Congrats, Doctor!

CVPR Workshop and Presentations

Turn Your CV Project into an App

Women in Computer Vision

Deep Learning Research

Upcoming Events

Dilbert

RSIP VISION
Global Leader in Computer Vision and Deep Learning

Computer Vision News

Editor:
Ralph Anzarouth

Engineering Editors:
Marica Muffoletto
Ioannis Valasakis

Head Designer:
Rotem Sahar

Publisher:
RSIP Vision
Contact us
Free subscription
Read previous magazines

Follow us:

Dear reader,

June has been a very busy month for our community, with a new and compelling edition of **CVPR**. Once again, **Virtual CVPR triumphed**. Scientists from around the world presented an impressive **1,500+ accepted papers** online.

As usual, **RSIP Vision** and its magazine, **Computer Vision News**, joined the party. We partnered with CVPR for the 6th consecutive year to publish **CVPR Daily** throughout the conference. In this July issue of Computer Vision News, we showcase the best of the best from CVPR 2021. Our **BEST OF CVPR** section features just a tiny portion of the event, but you'll discover some of its biggest highlights and share in our most memorable moments from a truly captivating week.

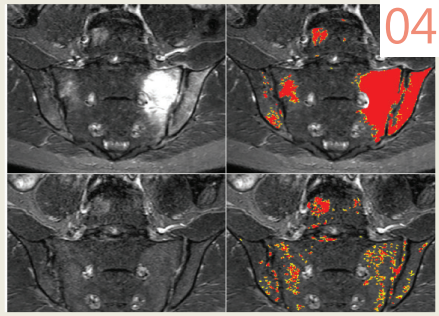
Among these highlights, we have our exclusive review of a fascinating paper by **Piotr Dollár** and **Facebook AI Research (FAIR)**. His innovative paper analyzes strategies for **scaling convolutional neural networks to larger sizes**, such that they are both fast and accurate. Don't miss this outstanding CVPR moment!
Away from CVPR, there is plenty more to interest you in this edition of Computer Vision News, so dive right in because our editors have prepared another exciting magazine for you!

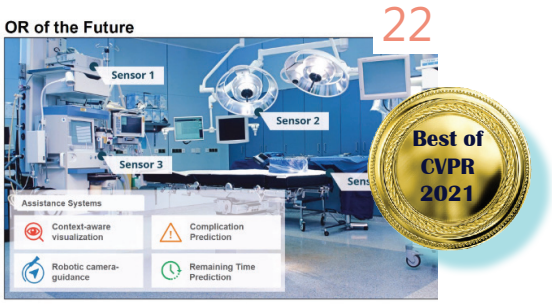
Enjoy the reading and subscribe for free!

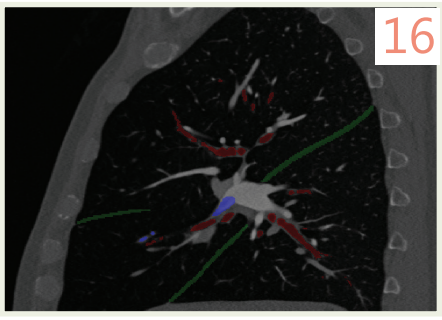
Ralph Anzarouth
Editor, **Computer Vision News**
Marketing Manager, **RSIP Vision**

Just a quick note to let you know that today's "CVPR Daily" is fantastic! Thank you for all your efforts! Really amazing work.

Michael Brown
General co-Chair, **CVPR 2021**
Professor, **York University**







Towards Deep Learning-assisted Quantification of Inflammation in Spondyloarthritis: Intensity-based Lesion Segmentation

IOANNIS VALASAKIS, KING'S COLLEGE LONDON
@WIZOFE

The research of the month is: Towards Deep Learning-assisted Quantification of Inflammation in Spondyloarthritis: Intensity-based Lesion Segmentation. It is currently pre-published on ArXiv by authors Carolyna Hepburn, Hui Zhang, Juan Eugenio Iglesias, Alexis Jones, Alan Bainbridge, Timothy JP Bray and Margaret A Hall-Craggs.

A new article for a new month! Happy fourth of July to our US readers

This month's article diverts from the previous ones, concentrating more on the quantification side of medical imaging and specifically the quantification of inflammation in spondyloarthritis. It's not a surprise anymore that some techniques of deep learning will be used! Read on to explore more of this very interesting topic.

Axial Spondyloarthritis

Axial Spondyloarthritis (axSpA) is defined as a chronic, inflammatory rheumatic disease that affects primarly the axial skeleton, causing severe pain, stiffness and fatigue. The disease typically starts in early adulthood. The current clinical standard of detection is with a short-tau inversion recovery (STIR) magnetic resonance imaging (MRI). In order to evaluate and treat the disease, the need of identification and quantification of inflammation is crucial.

As counter-intuitive as it may seem, at the moment the only way to describe the inflammation in clinical practice is a visual assessment of the MRI with a verbal description that includes no numerical metric to allow quantification. There is clinical research on using semi-quantitative scoring to describe bone marrow edema (BME, the inflammation in the subchondral bone marrow), but those are limited. They take a very long time to be calculated as well and they only take information from very few slices.

The main approach of the visual evaluation is based on the intensity. As such and because of the difficulty to define subtle lesions depending on light, the STIR MRI interpretation is causing variability.
The main aim of this article is to explore an automated deep learning-based approach that would detect and quantify inflammation and is fast and reproducible.

Approach and data

The dataset for this experimental work comes from University College London (UCL) hospital for a total of 30 subjects. Of them, 16 were females and 14 were males. The aim of this dataset was to evaluate prediction and responsiveness using quantitative imaging biomarkers.

The protocol was MRI STIR and T1-weighted turbo spin echo sequences on a 3T Philips Ingenia scanner.

For previous studies of quantification, manual segmentation was performed. Two readers performed such segmentation of inflammatory lesions to use in the supervised deep learning approach. The main aim of this architecture is to provide a second opinion, such as an AI advisor for the improvement of the workflow and the reduction of the workload by filtering some of the cases.

In Figure 1 you can see the overall approach on the data flow, including the manual segmentation from the readers and you can read it in even more details in the original paper.

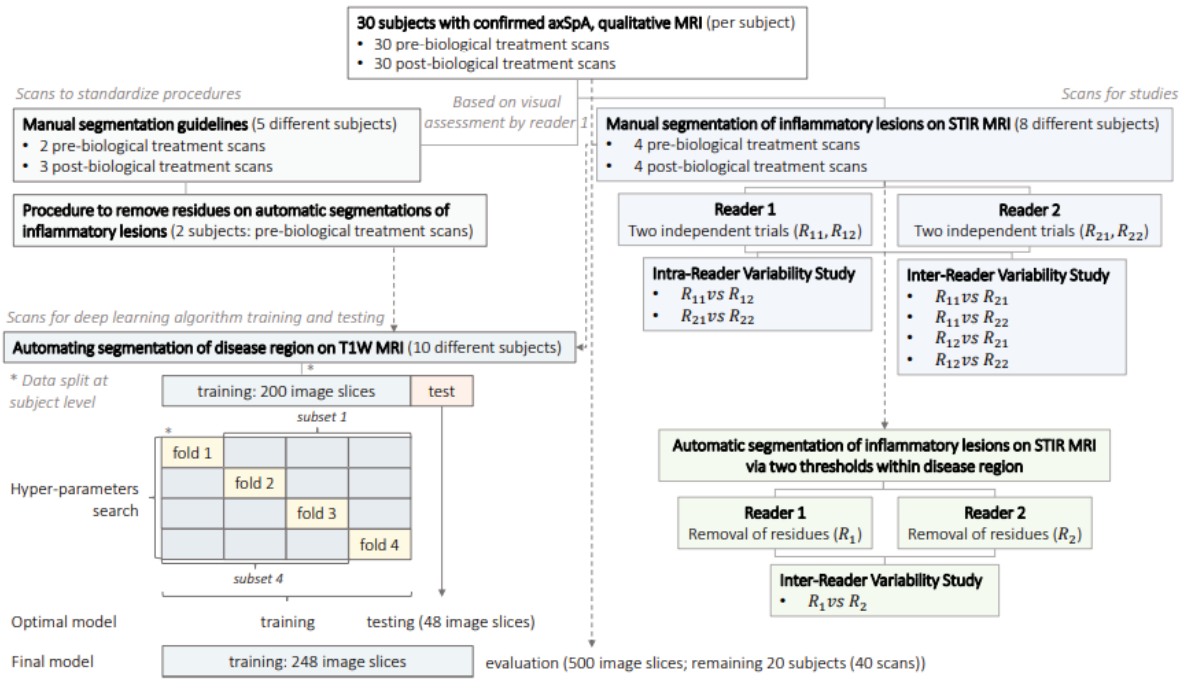


Figure 1. The data workflow and the scan availability for the creation of the AI model.

Deep learning model

Image slices from the T1-weighted MRI scans were used to train the model. Data were partitioned in sets with four-fold cross validation, categorized in sets for test, training, validation. The final model evaluation was done on the 20 subjects which means 500 image slices.

Heart of the architecture is a 2D U-Net trained on mini-batches by optimizing the binary cross entropy loss using the Adam optimizer. To allow for quicker convergence, batch normalization was used.
The model is visualized in Figure 2 below.

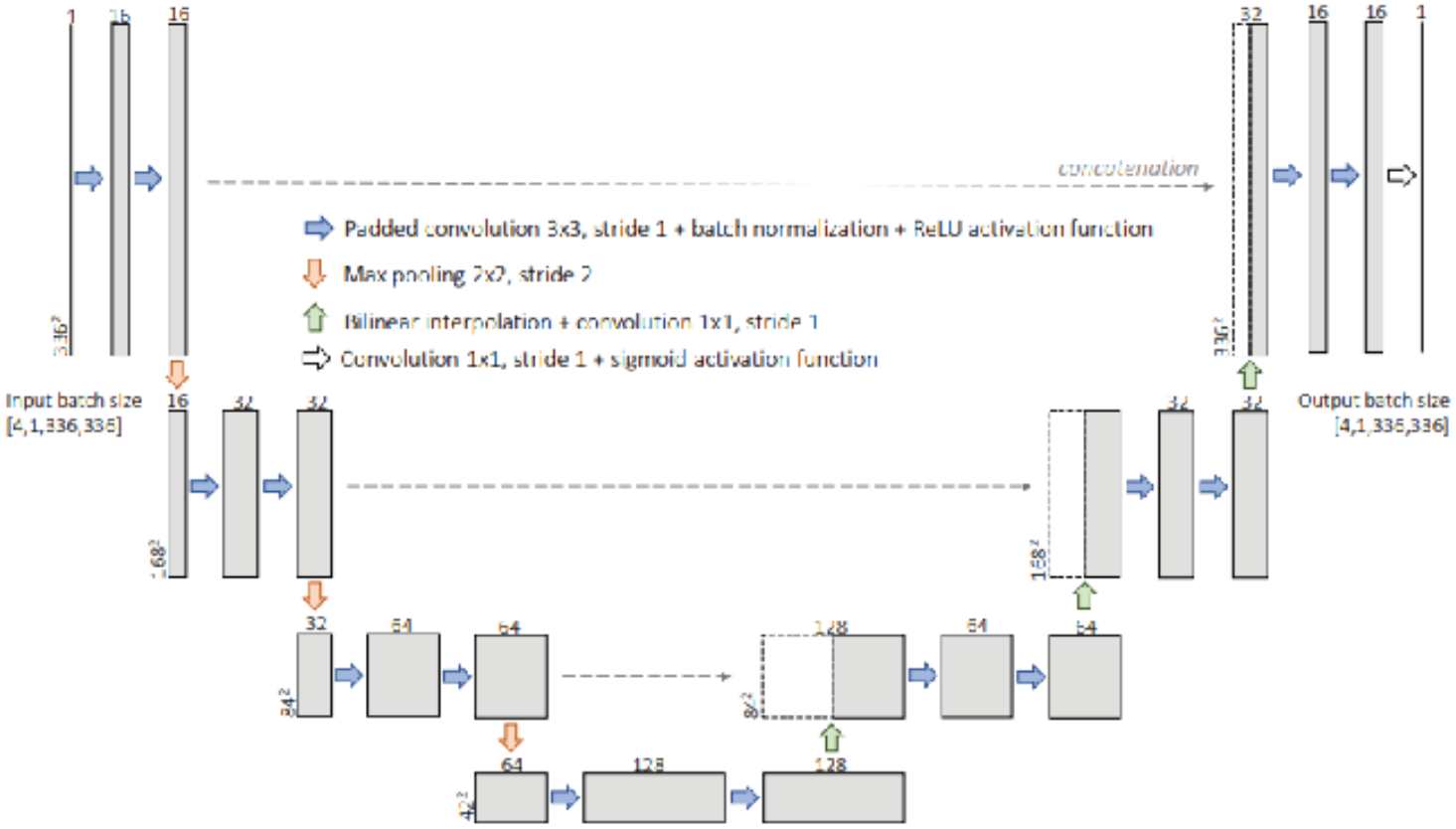


Figure 2: The architecture of the 2D network

A differentiable generalization of the dice score was used to evaluate network exploring the similarity between two binary segmentation volumes.

How does it perform?

In Figure 3 you can see 3D rendering of manual segmentation trials from both readers. What one could observe is the degree of fragmentation as well as how the location of the lesions differs. Readers would disagree on those locations which affected the dice score.

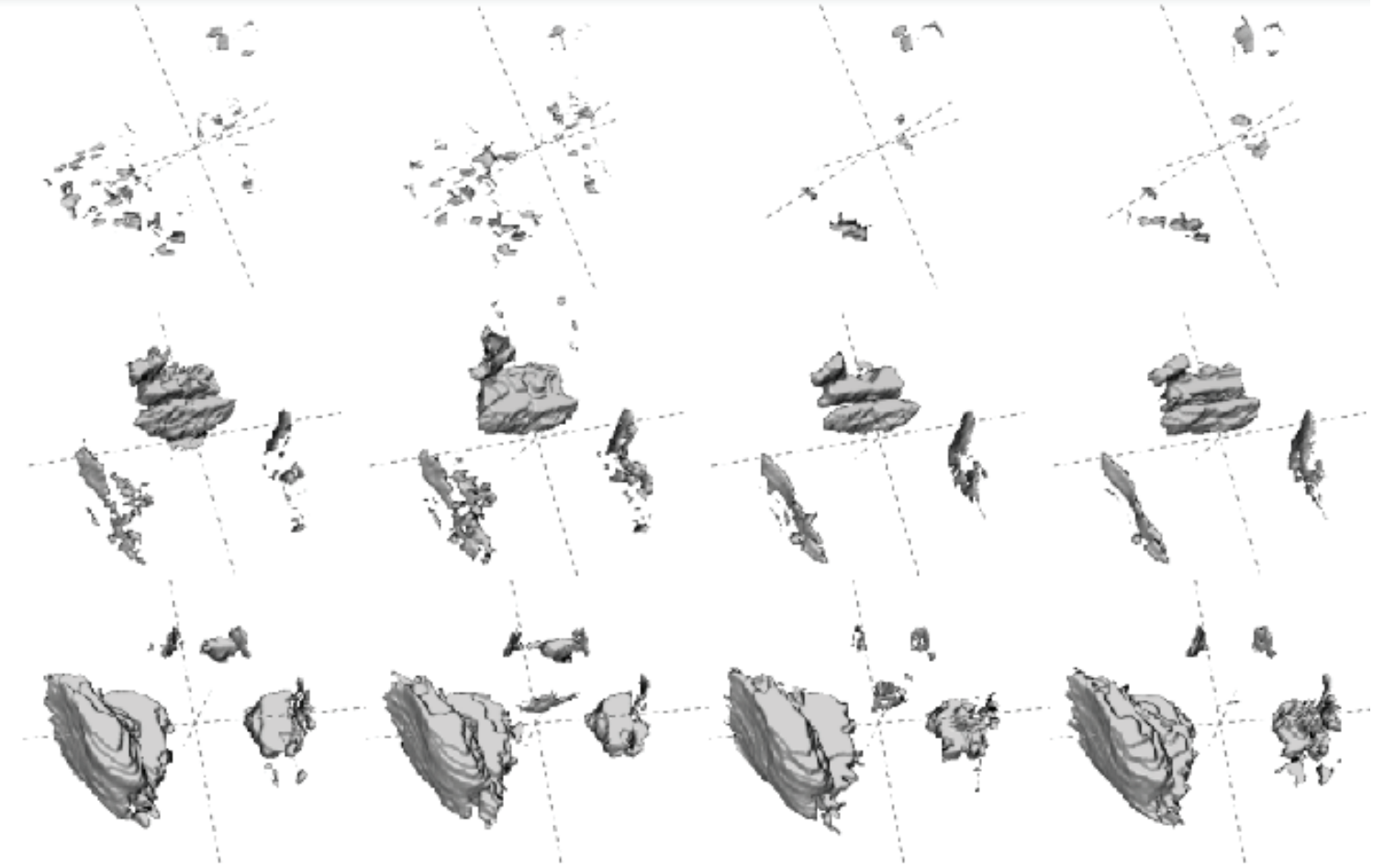


Figure 3: Here is the three-dimensional rendering of the manual segmentation trials from the two readers for three chosen subjects.

To estimate the optimal parameters of the network and perform the automatic (deep learning-based) evaluations, a cross-parameter grid search was performed, by varying hyper-parameters such as the network depth, number of epochs and kernels. There was no overfitting for the optimal model, but some fluctuations in performance for the fourth validation fold were seen.

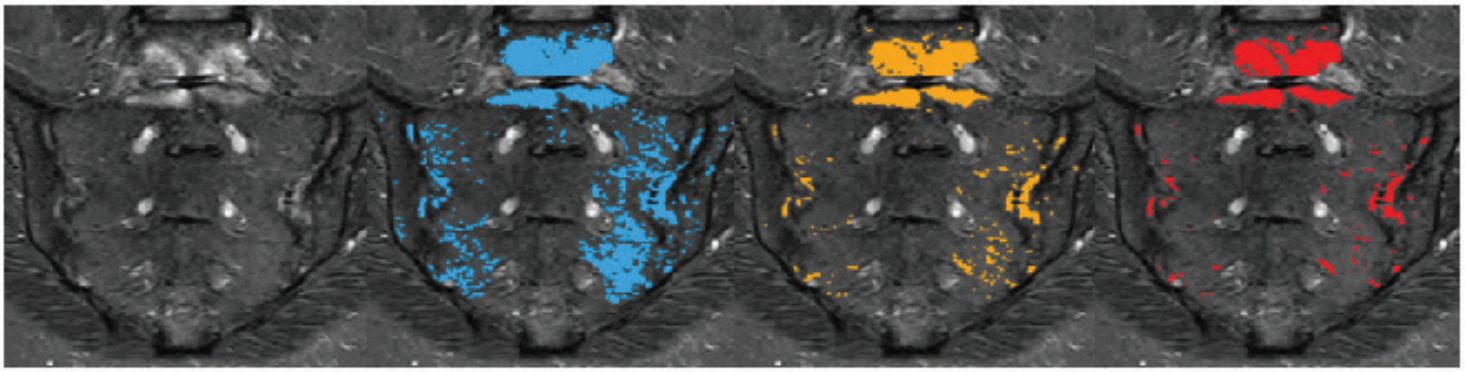


Figure 4: Oblique coronal plane with a super-imposed automatic segmentation using three different thresholds (Qu + 1.5 IQR, Qu + 2.5 IQR, maximum threshold).

Based on the results acquired from this model, a semi-automated workflow was developed to correct for the bias of readers. This semi-automated approach may create a few problems. By adjusting the threshold value (from a human reader) variability will be seen, which may even be stronger than the one observed! In Figure 4 a visualization can be seen of the super-imposed segmentation with different thresholds. In Figure 5, you can see the pre and post-biological treatment and the corrected super-imposed segmentation by the second reader (plus the automated algorithm).

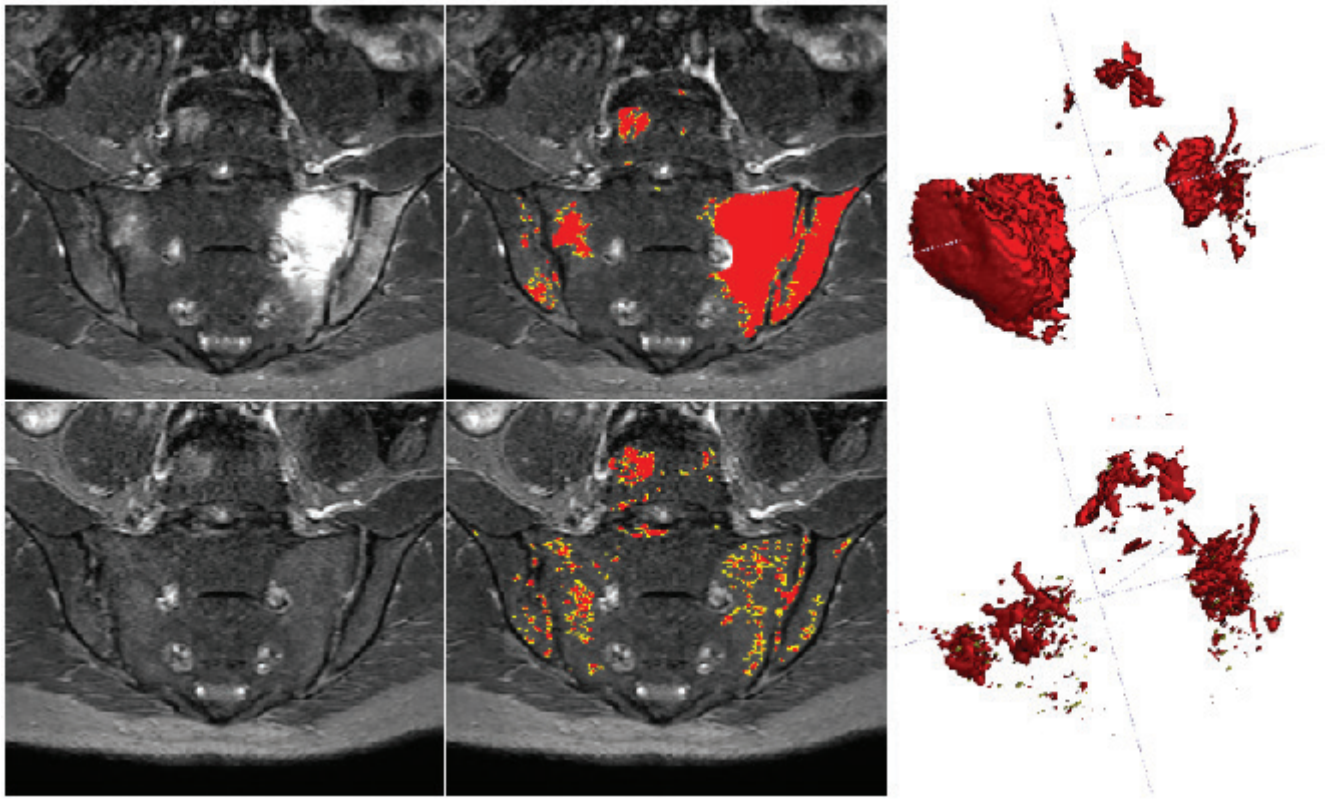


Figure 5: Here are the MRI slices with the STIR using pre, post-biological treatment (left/top respectively) and with super-imposed corrected by the second reader automatic segmentations

Even though the deep learning network can provide a correction, it does not eliminate the variability. This is partially due to the variation between abnormal and normal water content due to the cellularity/expansion of the extracellular space but also from the variation between the water content of (e.g. from sclerosis).

Wrapping up!

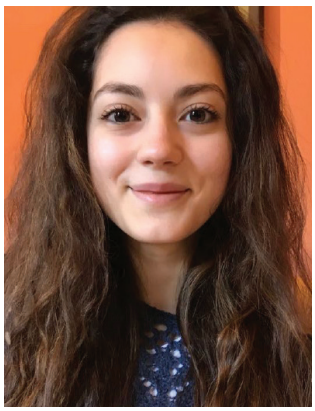
Thanks for staying through this article! I would like to thank C. Hepburn and the authors for writing this great paper ☺ Stay tuned for more news and many interesting articles from all the colleagues!☺

Are you attending the Medical Imaging and Deep Learning conference? It will be held virtually this month, on July 7-9.

MIDL
Lübeck 2021

Whether you are attending or not, don't miss the BEST OF MIDL in Computer Vision News of August! It will include some of the best scientific moments of the conference. We also plan to review the work of the Best Paper award winners!

How to easily turn your Computer Vision project into an app



by Marica Muffoletto

Dear readers, have you ever wished to make an app out of a very cool project you have been working on and share it with your friends after work? Or maybe thought about showing your results to your team with something more interesting and less casual than random graphs and images saved on your local machine, but maybe through a nice-looking interactive link? Well, it's way easier than you can imagine now with the open-source Python library **Streamlit**. This works just as any other Python library: it can be installed through pip and allows you to deploy your custom web app in just a few minutes. The only requirement is to have Python 3.6 - Python 3.8 on your machine, and then you will be good to go!

Streamlit apps are Python scripts that run from top to bottom. They can be accessed through a sharable link which every time is clicked will execute the script again. It also offers a wide range of widgets that the user can interact with and, once this is done, the script is re-executed and updated very fast.

We will see how to use it to create a little app which shows two common computer vision applications: segmentation and object detection. The first will run using a simple K-means algorithm while the second uses pre-downloaded weights and model of the notorious YOLO algorithm.

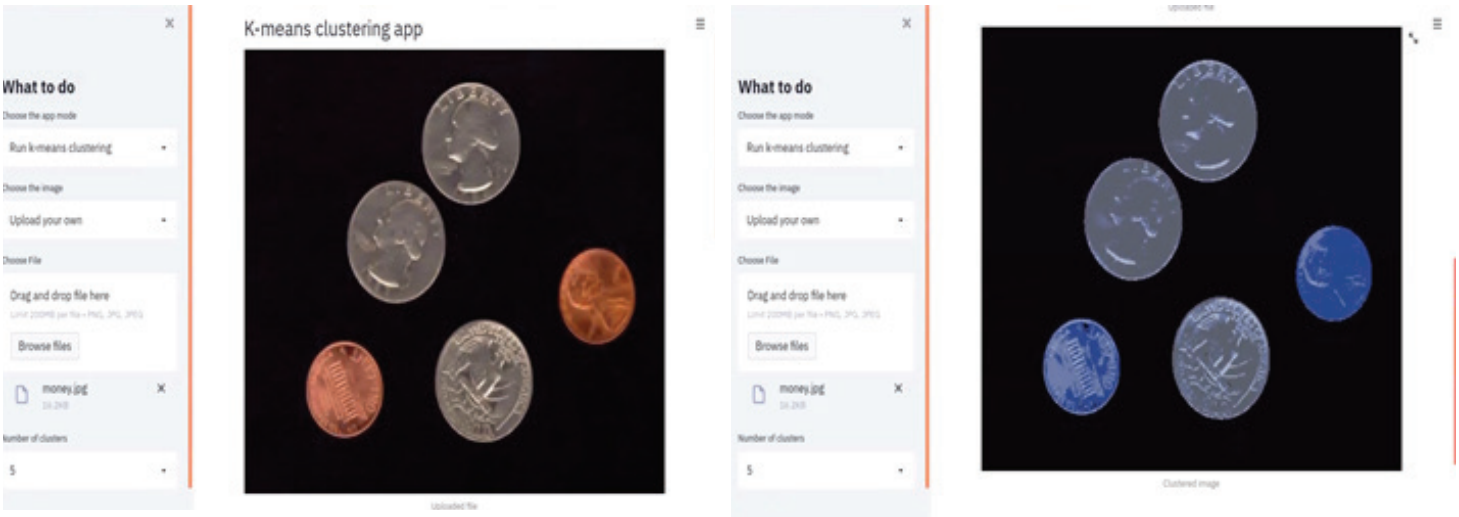
Let's start with designing our app for the first application (run_kmeans) through the code below.

```
import streamlit as st
import altair as alt
import pandas as pd
import numpy as np
import os
import urllib
import cv2
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import streamlit as st
import random
from PIL import Image
```

```
def upload_image(selection):
    if selection == "Upload your own":
        filename = st.sidebar.file_uploader(
            "Choose File", type=["png", "jpg", "jpeg"])
    else:
        filename = "exampleyolo.jpeg"
    return filename

def main():

    st.sidebar.title("What to do")
    app_mode = st.sidebar.selectbox("Choose the app mode",
                                    ["Run k-means clustering", "Run object
detection"])

    image_selection = st.sidebar.selectbox("Choose the image",
                                           ["Upload your own", "Use
example"])

    if app_mode == "Run k-means clustering":
        image_filename = upload_image(image_selection)
        run_kmeans(image_filename)
    elif app_mode == "Run object detection":
        image_filename = upload_image(image_selection)
        run_detection(image_filename)

# Cached function that returns a mutable object with a random number in the
range 0-100

@st.cache(allow_output_mutation=True)
def seed():
    return {'seed': random.randint(0, 100)}  # Mutable (dict)

# This is the function that runs the k-means algorithm itself, which appears
when the user selects "Run k-means clustering".

def run_kmeans(file_uploaded):

    st.header("K-means clustering app")

    if file_uploaded is not None:
        image = Image.open(file_uploaded)
        st.image(image, caption='Uploaded file', use_column_width=True)
        cvimage = np.array(image)
        # convert to RGB
        cvimage = cv2.cvtColor(cvimage, cv2.COLOR_BGR2RGB)
        # reshape the image to a 2D array of pixels and 3 color values (RGB)
        pixel_values = cvimage.reshape((-1, 3))
        # convert to float
        pixel_values = np.float32(pixel_values)
        # define stopping crpiteria
        criteria = (cv2.TERM_CRITERIA_EPS +
                    cv2.TERM_CRITERIA_MAX_ITER, 100, 0.2)
        # choose number of clusters (K)
        k = st.sidebar.selectbox('Number of clusters', range(1, 10))
        _, labels, (centers) = cv2.kmeans(pixel_values, k, None,
                                          criteria, 10,
```

```
cv2.KMEANS_RANDOM_CENTERS)
        # convert back to 8 bit values
        centers = np.uint8(centers)
        # flatten the labels array
        labels = labels.flatten()
        # convert all pixels to the color of the centroids
        segmented_image = centers[labels.flatten()]
        # reshape back to the original image dimension
        segmented_image = segmented_image.reshape(cvimage.shape)
        # show the segmented image
        st.image(segmented_image, caption='Clustered image',
use_column_width=True)


# This is the function that runs the YOLO algorithm, which appears when the
user selects "Run object detection".

def run_detection(file_uploaded):

    st.header("Run detection app")

if __name__ == "__main__":
    main()
```

Now, to try out the result of a freshly made app one needs to type: ***streamlit run filename.py*** on the command line. This will print a link to the streamlit app! That's how my app looks like at this stage.

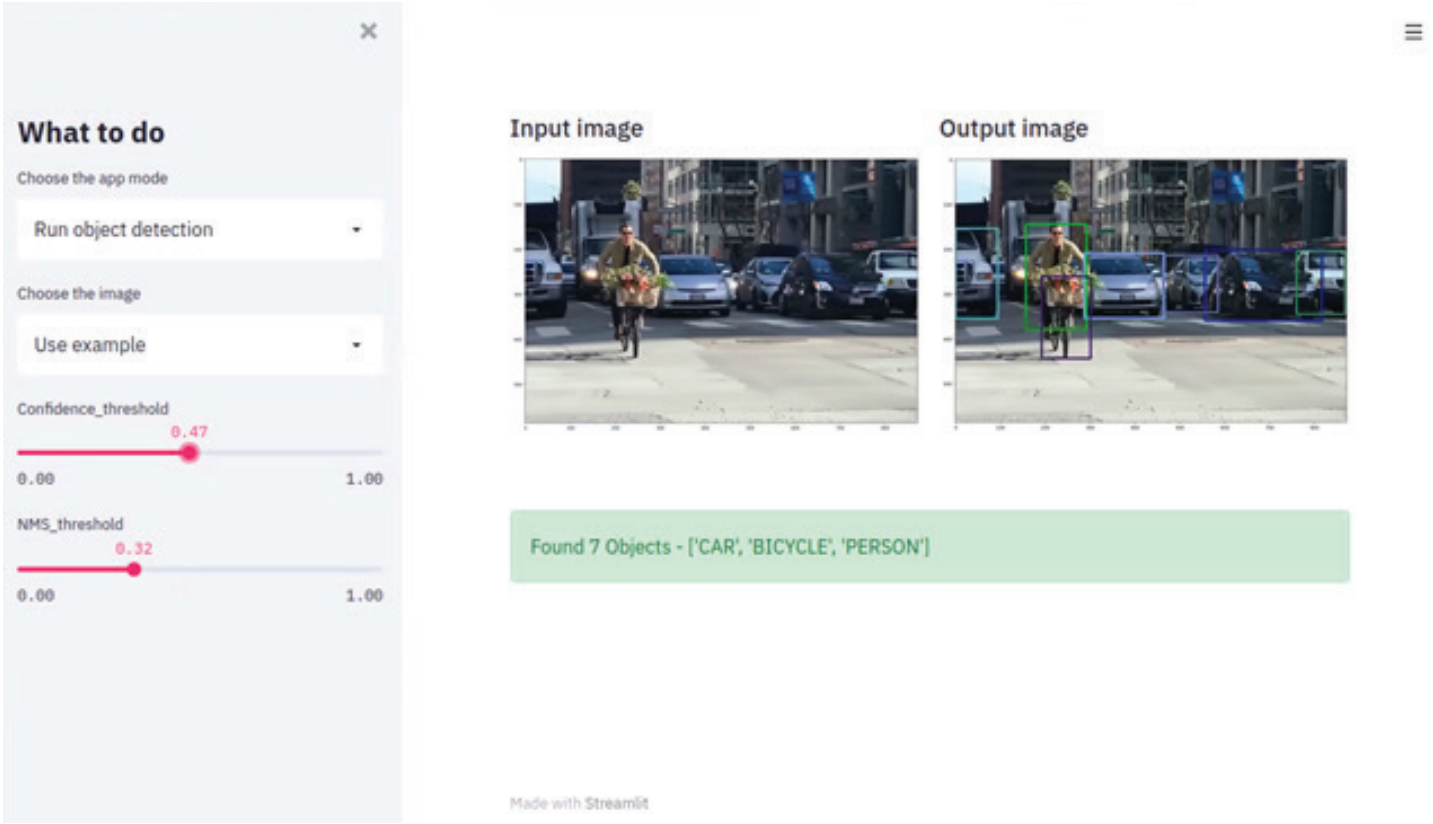


The following part of the code is dedicated to filling in the run_detection function which will show as a separate section in the app (just scroll down to see the result!).

```
def run_detection(file_uploaded):

    st.header("Run detection app")

    image = Image.open(file_uploaded)
    st.set_option('deprecation.showPyplotGlobalUse', False)
    column1, column2 = st.beta_columns(2)
    column1.subheader("Input image")
```

```
    st.text("")
    # Display the input image using matplotlib
    plt.figure(figsize=(16, 16))
    plt.imshow(image)
    column1.pyplot(use_column_width=True)

    neural_net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
    labels = []  # Initialize an array to store output labels
    with open("coco.names", "r") as file:
        labels = [line.strip() for line in file.readlines()]
    names_of_layer = neural_net.getLayerNames()
    output_layers = [names_of_layer[i[0]-1] for i in
neural_net.getUnconnectedOutLayers()]
    colors = np.random.uniform(0, 255, size=(len(labels), 3))
    newImage = np.array(image.convert('RGB'))
    img = cv2.cvtColor(newImage, 1)
    height, width, channels = img.shape
    # Convert the images into blobs
    blob = cv2.dnn.blobFromImage(
        img, 0.00391, (416, 416), (0, 0, 0), True, crop=False)
    neural_net.setInput(blob)  # Feed the model with blobs as the input
    outputs = neural_net.forward(output_layers)
    classID = []
    confidences = []
    boxes = []
    # Add sliders for confidence threshold and NMS threshold in the sidebar
    score_threshold = st.sidebar.slider("Confidence_threshold",
                                        0.00, 1.00, 0.5, 0.01)
    nms_threshold = st.sidebar.slider("NMS_threshold", 0.00, 1.00, 0.5,
                                      0.01)
    # Localise detected objects in the image
    for op in outputs:
        for detection in op:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5:
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)  # centre of object
                w = int(detection[2] * width)
                h = int(detection[3] * height)
                # Calculate coordinates of bounding box
                x = int(center_x - w / 2)
                y = int(center_y - h/2)
                # Organize the detected objects in an array
                boxes.append([x, y, w, h])
                confidences.append(float(confidence))
                classID.append(class_id)

    indexes = cv2.dnn.NMSBoxes(boxes,confidences, score_threshold,
nms_threshold)
    # Assign color to different objects
    items = []
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = str.upper((labels[classID[i]]))
            color = colors[i]
            cv2.rectangle(img, (x, y), (x+w, y+h), color, 3)
            items.append(label)
```

```
    st.text("")
    column2.subheader("Output image")
    st.text("")
    # Plot the output image with detected objects using matplotlib
    plt.figure(figsize=(15, 15))
    plt.imshow(img)  # show the figure
    column2.pyplot(use_column_width=True)
    # Print how many objects are detected
    if len(indexes) > 1:
        st.success("Found {} Objects - {}".format(len(indexes),[item for item
in set(items)]))
    else:
        st.success("Found {} Object - {}".format(len(indexes),[item for item
in set(items)]))
```



Once you are satisfied with the result, you can finally decide to deploy your app and make it public! This is a very smooth process too.

First, you will need to register on this website and then add the Streamlit app (yes that's just a tiny Python script!) to a public GitHub repo, add a requirements file to manage external dependences (.txt file for pip, .yml file for conda). Now if you log in to share.streamlit.io with your GitHub account email, you will be able to deploy the app by clicking on "New app" and fill in the required info. Your app is now accessible through a link! This will usually contain user and repo name linked to the

github account and it can be shared. Enjoy surprising your friends and colleagues with your new little gem!
As from this review, Streamlit is very intuitive. Its creators offer a wide community you can join, and made it super easy to explore it through the Streamlit docs or also visiting Awesome Streamlit docs.
Wishing you the best of luck and fun with this fantastic library! ☺

Pulmonary Embolism Detection Using AI

Pulmonary Embolism (PE) is a life-threatening condition with a mortality rate of up-to 30%, where an embolus blocks one of the pulmonary arteries. When the thrombus originates in other blood vessels (usually deep vein thrombosis) it is defined as acute PE. Occasionally the clot develops over time and inseparable from the vessel wall, slowly increasing the artery blockage. This is defined as **chronic Pulmonary Embolism**.

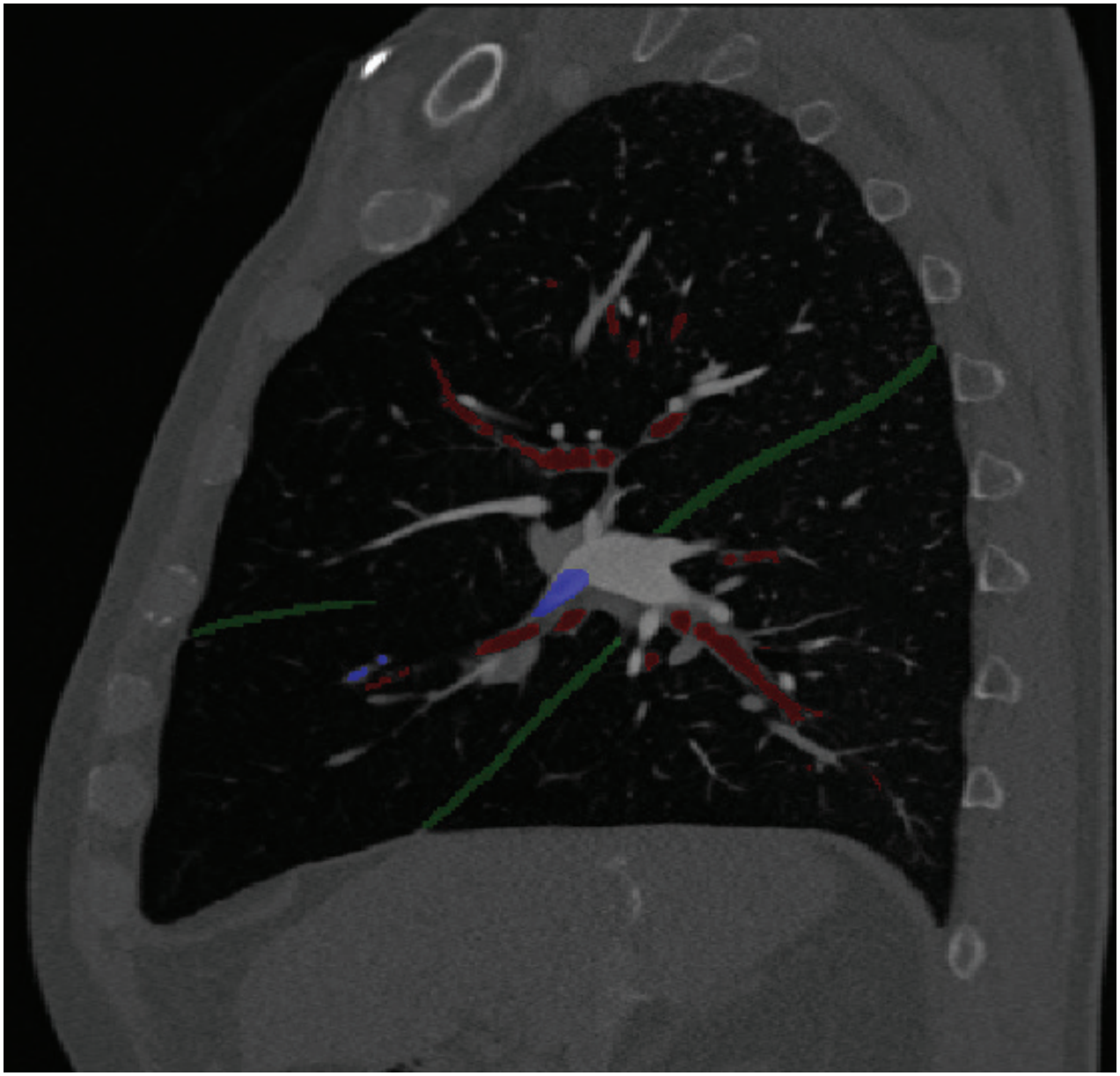
PE treatment needs to be immediate and accurate, especially in acute PE. Common diagnostical procedure requires **CT Pulmonary Angiography (CTPA)**, due to its availability, speed, and accuracy. A high-resolution 3D scan of the pulmonary arteries assists in PE detection. However, precise detection of the emboli can be challenging – it is difficult to track pulmonary arteries throughout the scan, and errors are often made.

RSIP Vision has developed an AI-based Airways Segmentation suite. This system robustly segments lungs, lobes, nodules, airways, and pulmonary blood vessels from CT scans. Pulmonary arteries and airways run alongside each other in hierarchical bifurcations from lobes to secondary pulmonary lobule, suggesting that knowledge of airways structure can assist in blood vessel segmentation and emboli detection.

The above introduction emphasizes the need for an **automatic PE detection tool**. Combining Artificial Intelligence and classic computer vision algorithms can efficiently and accurately detect the location of the emboli based on the CTPA scan. This system will maximize the segmentation accuracy of the pulmonary arteries as they will stand out compared to the rest of the pulmonary structures, making emboli detection easier.

The main challenge of segmentation from CTPA is **overcoming image artifacts** originating from movement (breathing), implants, human error, and anatomical differences. **Advanced neural networks** have proven capabilities of achieving a trustworthy solution despite poor image quality.

Apart from detecting the vessel filling defect, additional clinical data is necessary for treatment selection. Enlargement of the right atrium or ventricle, as well as dilation of the pulmonary arteries influence the physician's decision-making. Segmentation of the cardiac chambers and proximal pulmonary arteries is feasible from the CTPA using similar **deep learning techniques**, and automatic detection of abnormalities can be calculated and provided as an input for clinical decision-making.



Green - Fissures Red - Airways Blue - Pulmonary Embolism

Overall, this process combines several AI tools: **airway segmentation, pulmonary blood vessel segmentation, cardiac chambers segmentation, automatic blood vessel blockage detection, and cardiac chamber and blood vessels dilation calculator**. The single input for this system is a CTPA scan, whereas the output is precise embolus position and additional risk factors. Instead of a radiologist manually assessing all these, within seconds of the scan the diagnosis is available.

AI, specifically **deep learning**, has tremendous potential in speeding diagnostical procedures. Due to PE's high mortality rate and fast deterioration pace, it requires **immediate diagnosis and treatment**. Incorporating such a capability into PE healthcare can significantly improve the outcomes by drastically reducing manual labor time.

Fast and Accurate Model Scaling

Piotr Dollár is a research director at Facebook AI Research (FAIR).

His innovative paper analyzes strategies for scaling convolutional neural networks to larger scales, such that they are both fast and accurate.

There are various **strategies to scale a network**, like scaling a ResNet-50 into a ResNet-101 by doubling the depth, but what if you want very high-capacity networks and you want them to be reasonably fast? This work looks at what makes these models both accurate and *fast* when scaled and provides a paradigm for scaling networks in an efficient way.

Until now, people primarily looked at how scaling strategies affect flops – floating point operations – and the accuracy of a model. The problem with this is that flops are not very predictive of runtime on modern accelerators (GPU, TPU). **Modern accelerators are very memory-bound**, so often what really matters is the memory these networks use.

This work uses a notion called **activations**, which is the size of the output tensors of convolutional layers in CNNs. **Activations – which are roughly correlated to memory – are an analytical measure which you calculate offline about a network and are very predictive of runtime**. Piotr and his team designed activations-aware scaling strategies so that when you scale a network, say to double its flops, you are also cognizant of how the activations change. Therefore, these networks are much faster on modern accelerators.

"*What's exciting about this work is that one would think there's* **a trade-off *between speed and accuracy*** *as you get to these bigger models,*" Piotr tells us. "*In previous work, the very big models tended to be quite slow, or they tended to be inaccurate. But you can get the best of both worlds – models that are very big and both reasonably fast and accurate.* ***It's a win-win!***"

The team has been working on **model design and understanding** for some time

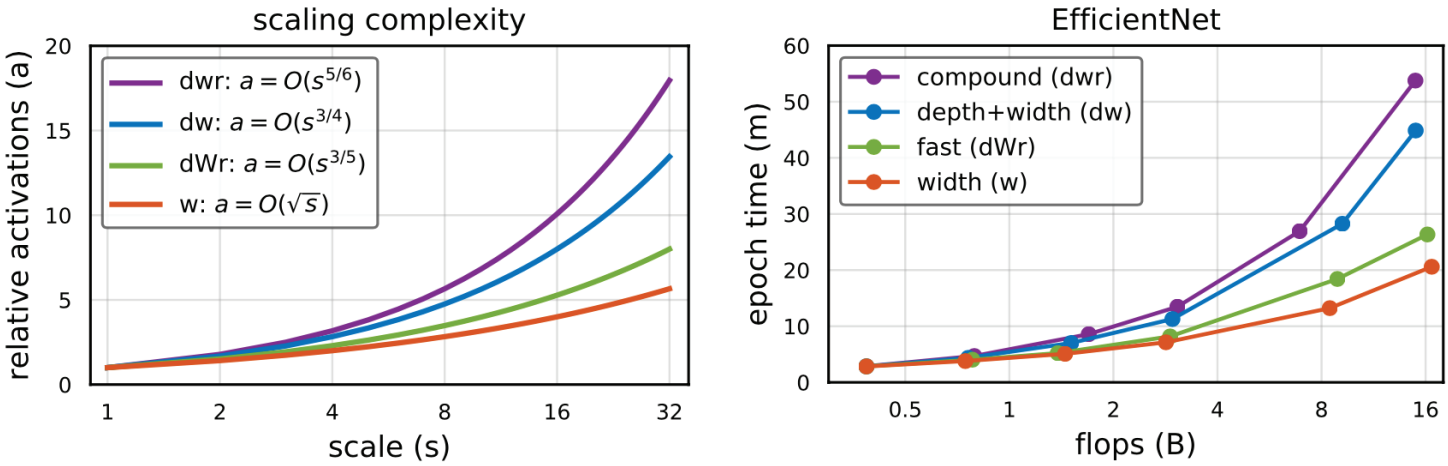


Figure 1: An analysis of four model scaling strategies: width scaling (*w*), in which only the width of a base model is scaled; compound scaling (*dwr*), in which the width, depth, and resolution are all scaled in roughly equal proportions; depth and width scaling (*dw*); and the proposed **fast compound scaling** (dWr), which emphasizes scaling primarily, but not only, the model width. The left plot shows theoretic behavior of activations w.r.t. different scaling strategies, the right shows the actual runtime of scaling a small network (EfficientNet-B0 in this case) using these scaling strategies. Observe that: (1) activations scale at asymptotically different rates for different scaling strategies, (2) activations are highly predictive of runtime (shape of left and right plots closely matches), and (3) fast compound scaling (dWr) is very fast (nearly as fast as width-only scaling which is the fastest scaling strategy). Furthermore, the accuracy of *dWr* scaling matches the accuracy of *dwr* scaling and easily outperforms *w* scaling (accuracy not shown). Hence fast compound scaling is a win-win: it is as accurate as compound scaling and nearly as fast as width only scaling.

now. The field has a proliferation of different model architectures that are effective or fast, but rather than just trying to create an effective network, they have been trying to understand the principles that make it effective. So, not just finding a fast or accurate model, but trying to understand that makes models effective and how to scale them. Similarly, at **CVPR 2020**, the team presented a paper called **Designing Network Design Spaces**, which explored principles for designing effective convolutional neural networks, rather than just looking for effective individual model instances.

"*In our community, people have been really pushing absolute numbers, and one key aspect of getting good numbers is the training recipe,*" Piotr explains. "*When you train a neural network, you can add all kinds of augmentations, regularizations, and so on, you tune the learning rate, the weight decay, and other hyperparameters, all this is essential to getting high numbers. And that is*

often hidden in a lot of papers, which makes it difficult to compare models and reproduce results. People often report really good numbers and attribute that to the model, but even though the model is good, it is the training recipe that is required to make those numbers great."

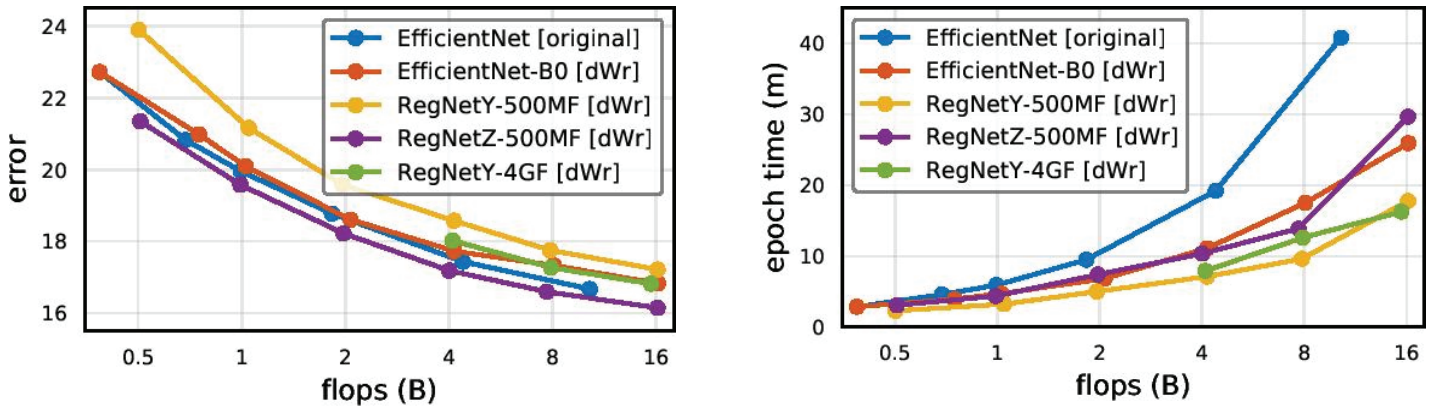


Figure 2: Fast scaling applied state-of-the-art CNNs at larger scales. Left: error vs flops for various CNNs; right: runtime versus flops. Fast scaling is effective for a number of different CNNs and results in as accurate but faster models than the previous best scaling strategies.

In this work, the team wanted to make sure the training recipe was easy to reproduce and not overly complex, but still gave good results. This was challenging because while there are very simple recipes one can use that are easy to reproduce, their absolute numbers are not very good. The team spent a lot of time perfecting a training recipe that was reproducible, worked for a variety of settings and networks, and that generalized to lots of models. Reproducibility is something the community is challenged by in general, so they wanted to make sure they got it right. This was orthogonal to the main goal but is a valuable secondary contribution of the work.

A general challenge for the computer vision community has been how to get to **a very large scale in terms of data and models**. What bigger models should they use? **CNNs or ViTs** (vision transformers)? And **self-supervised learning or supervised datasets?** Piotr's work aims to address one aspect of the puzzle, which is how to scale CNNs to a very large scale, but many challenges and open questions remain.

"*I think our community is still in some sense lagging the **NLP community**,*" Piotr tells us. "*In NLP, self-supervised learning, things like BERT and so on, have been very successful, and their datasets and models are much larger as well. Our community is still trying to figure out what are the right giant models. Our work is on CNNs and now we're also looking at ViTs, which are a very promising model*

as well especially at scale. But we don't have great large-scale supervised datasets. ***Unsupervised learning*** *has seen amazing progress – for example Kaiming He has been doing amazing work in this field, including at CVPR this year – but it's still far away from supervised learning, unlike in NLP where unsupervised learning is the default."*

As such an experienced and respected scholar in the community, how did it feel to be first author again on such an important new paper?

"*I actually got to code, which was a privilege for me because I don't often get to do it anymore!*" Piotr laughs. "***This was a paper I coded and did all the experiments primarily myself****. I think it's fun for a more senior person to be able to get more hands-on."*

And as such an eminent member of the community, we asked Piotr if he had any words for first-time CVPR participants who are missing out on the in-person experience this year.

"*We're all in the same boat,*" he responds. "*We're all figuring this out. Nearly all our conference experiences are in an in person setting, so I think this puts us on a level playing field. We're all attending these conferences for the first time in a virtual way.*"

Piotr works closely with many of the top scholars in the community – **Kaiming He**, **Ross Girshick**, **Georgia Gkioxari** – so we can't let him go without asking how it feels to work alongside such talent?

"*I love that question!*" he smiles. "*It's incredible and humbling to work with them. A real privilege!* ***Georgia is incredible and there's some really exciting news for Georgia!*** **[Check on page 29 of this mag]** ***Ross and Kaiming both won the PAMI Young Researcher Award a few years ago****. Each of us has slightly different research agendas, but there's a common thread, which is their* ***research excellence****. Nowadays, it's very easy to just want to publish a flashy result, but they're all so dedicated to* ***the integrity of their work****. I think that's what has allowed them to be so successful over the long term."*

Finally, Piotr also has some very kind words for us:

"*It's great that you're featuring our work. I really appreciate it. I have been following Computer Vision News for years and I enjoy it very much!*"

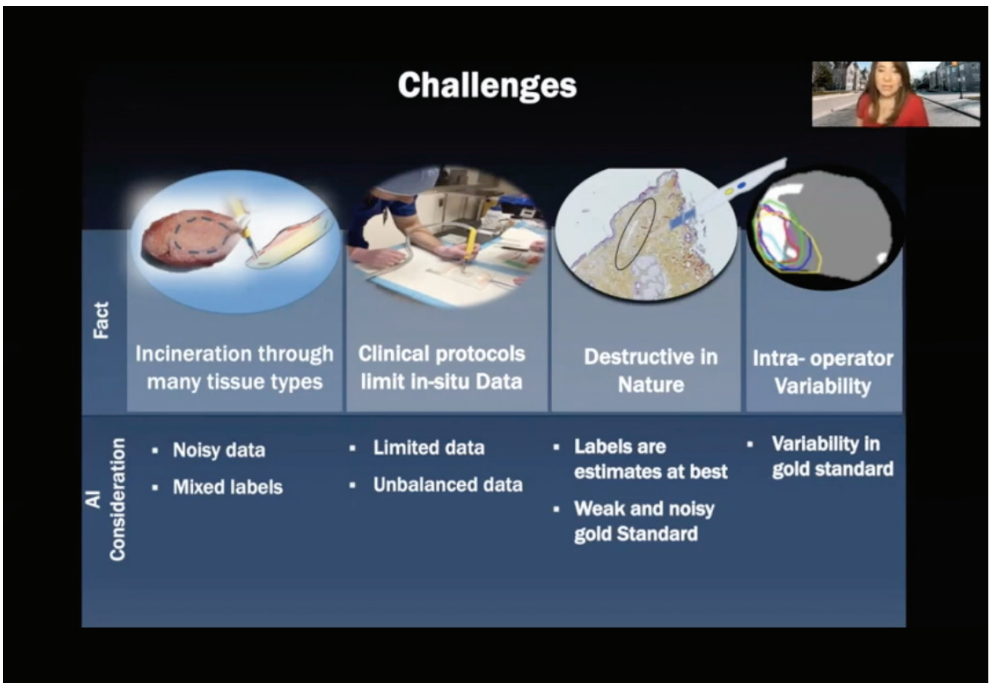
by Qi Dou (CUHK,qidou@cuhk.edu.hk)

Qi Dou

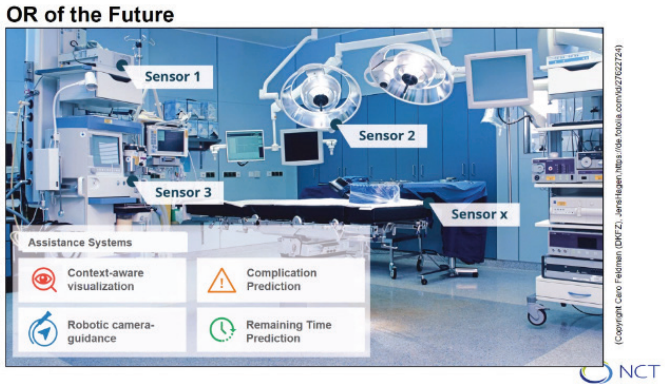
The 8th edition of the CVPR Medical Computer Vision workshop was held virtually on June 20, 2021. The workshop was organized by Qi Dou (The Chinese University of Hong Kong), Vasileios Belagiannis (Universität Ulm, Germany), Le Lu (PAII Inc., USA), Nicolas Padoy (University of Strasbourg, France), Lena Maier-Hein (German Cancer Research Center), and Tal Arbel (McGill University, Canada). The MCV workshop aims to bring closer the medical image computing and computer-assisted intervention community and the computer vision community in general, to provide a dedicated forum for idea exchanges, potential new collaborative and interdisciplinary efforts, and brainstorming new machine learning and computer vision applications in healthcare.

Improved from last year, the workshop offers a larger number of speakers up to 16 invited keynote presentations covering cutting edge research topics from distinguished world-class leaders from both academia, industry and clinicians in the field. The talks were delivered either through live steaming or pre-recorded videos. After each session of 4 talks, there is a live Q&A panel where audience members asked the speakers questions over chat and got answers within zoom. A total of around 500 attendees joined the event, with extensive discussions on latest work in progress, overview of recent technological advances, outstanding clinical needs, remaining challenges and opportunities. The schedule and the recorded talks can be found here.

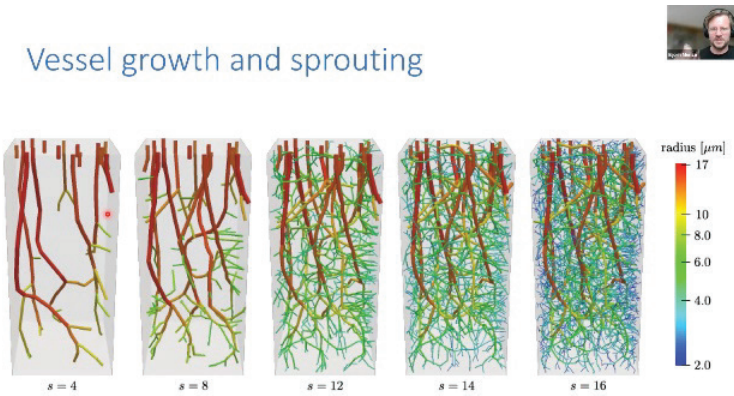
The workshop starts with an extraordinary talk from Nassir Navab, Professor at Technical University of Munich, presenting a great spectrum of ideas and decade of research works on computer vision techniques for computer assisted intervention. He described a family of exciting works on image processing, virtual reality, 3D reconstruction in ICU and surgical phase recognition and high-level understandings. Showing that computer vision gives more and more magic for surgical data science, he also shared constructive insights for future of the field. Parvin Mousavi, Professor at Queen's University, gave an excellent talk about AI for cancer surgery with insights on reimagining surgical oncology in the age of learning models, and showcased how her team's developed advanced intelligent image processing techniques can be used intraoperatively.



Gustavo Carneiro, Professor at University of Adelaide, gave a great talk that showcased inspiring works on anomaly detection and localization in medical images analysis with example use cases of colonoscopy images and brain images, involving generative adversarial, targeted self-supervised pre-training, and few-shot learning techniques. Stefanie Speidel, Professor at National Center for Tumor Diseases (NCT) Dresden, shared exciting works on video analysis for context-aware assistance in SurgeryOR4.0, with showcases of advanced techniques such as Sim2Real image and video translation and soft-tissue registration which are important topics towards OR of the Future. **Sotirios A. Tsaftaris**, Professor at University of Edinburgh, gave a fantastic talk emphasizing "Big AI" in Radiology, which has necessary ingredients of multiple tasks/inputs, generalization capability, less supervision and concept learning. The pathway to this exciting "Big AI" essentially relies on better disentangled representations which helps combat many challenges such as vendor variations and population difference. A broad group of challenges and opportunities including causality are also discussed with inspiring viewpoints. **Björn Menze**, Professor at University of Zurich, gave a wonderful talk entitled "On Vessels and Networks" presenting wonderful projects on high-resolution vascular data about vessel segmentations and prediction of vessels centerlines, with novel techniques of unsupervised learning, wise usage of synthetic data and topology awareness combined domain knowledge.



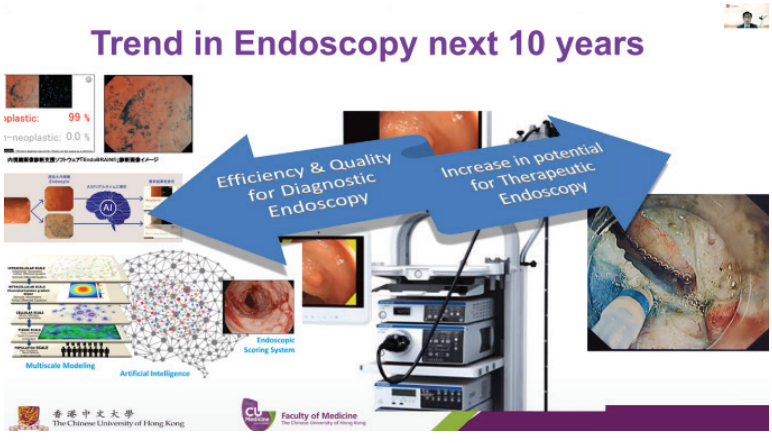
The workshop invited expert clinicians of oncology, surgery and

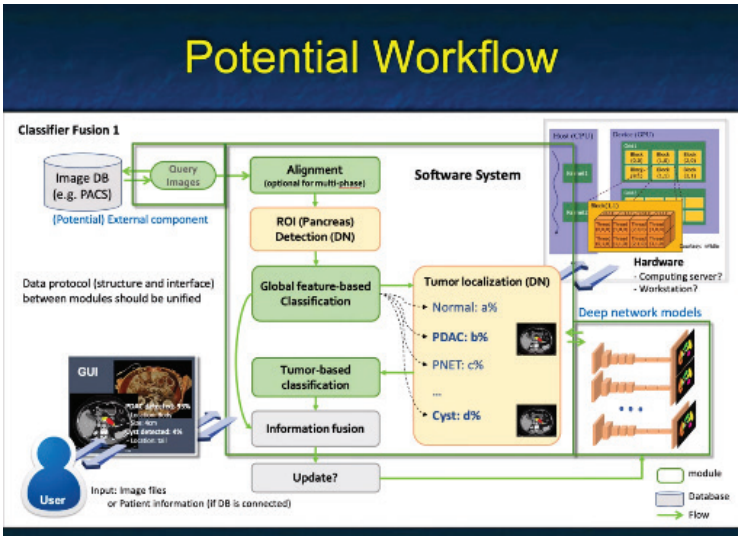


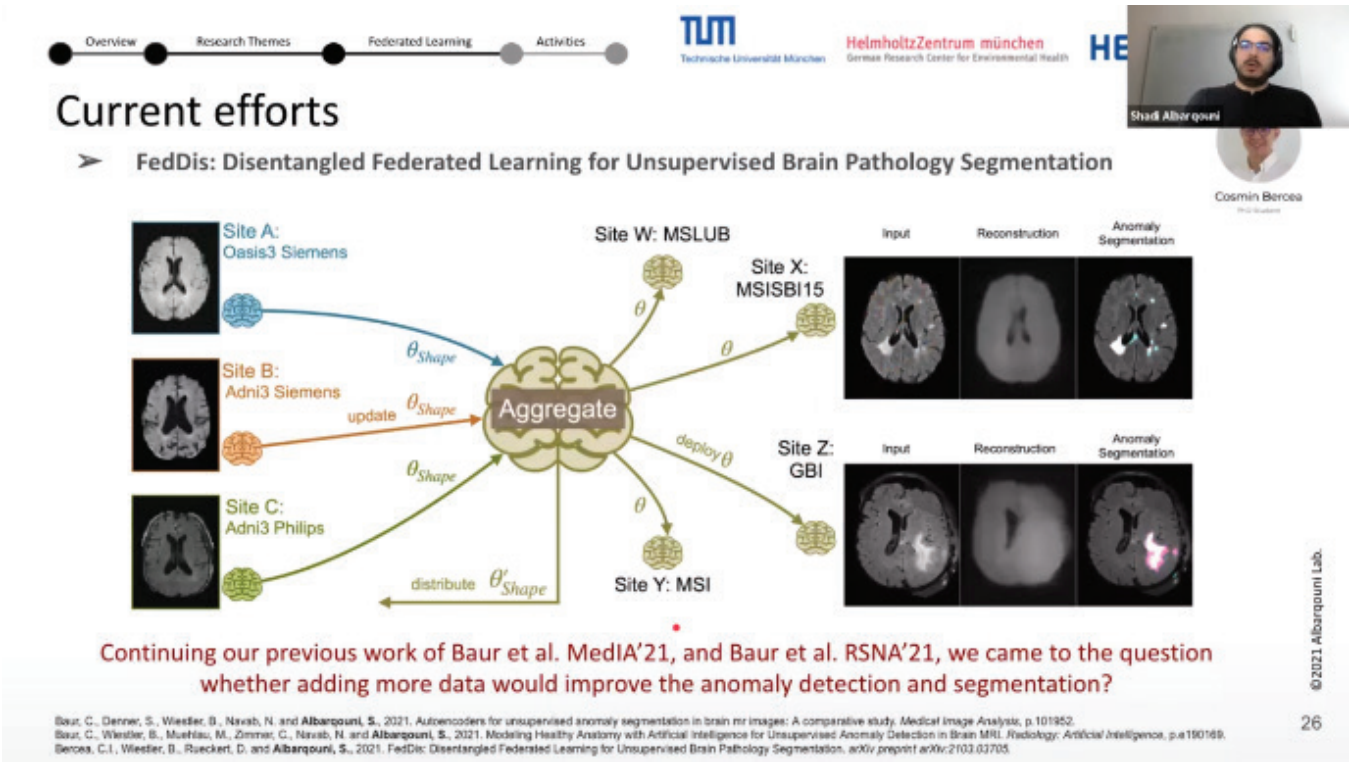
Iterative growth of a vascular tree

radiology, who have rich experience of AI applied to their respective context, and are optimistic about AI facilitating their clinical workflow in future. **Dorothy Tzu-Chen Yen**, Professor at Department of Nuclear Medicine and the Center for Advanced Molecular Imaging and Translation at Chang Gung University and Chang Gung Memorial Hospital, gave a wonderful talk on values of AI-based medical images, and presenting her recent series of new works on intelligent radiology image analysis from a clinician point of view. **Philip Wai-Yan Chiu**, Professor at Department of Surgery at The Chinese University of Hong Kong, gave an exciting talk that shared fantastic insights on the role of AI in endoscopy and minimally invasive surgery, in particular AI for standardization of endoscopic examination, lesion detection and characterization to meet the important clinical unmet need, and AI for enhancing therapeutic endoscopy for decision guidance, quality assurance and automation in procedure for the future. **Elliot Fishman**, Professor at Radiology, Oncology, Surgery and Urology at Johns Hopkins Hospital, gave a fantastic talk which showcased the role of AI in the earlier detection of pancreatic cancer through deep learning techniques and the combination of DL with radiomics features.

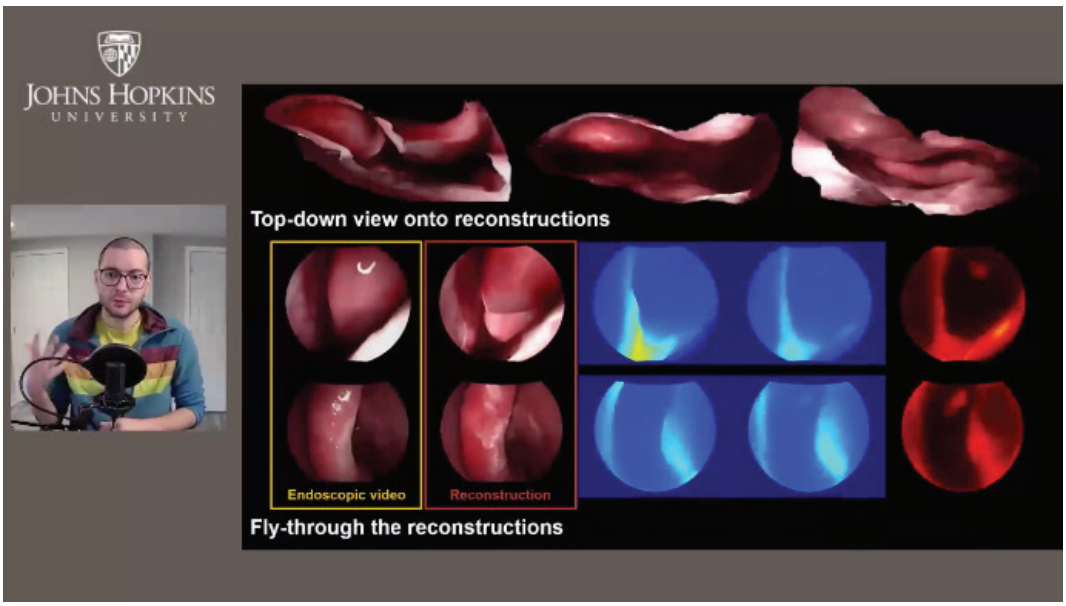
In the last two sessions, **Paul F. Jaeger**, German Cancer Research Center, give a fantastic talk presenting nnU-Net which has achieved many success stories as a powerful automated design of deep learning method for biomedical image segmentation and is published in Nature Methods. **Shadi Albarqouni**, Helmholtz Zentrum München, gave a wonderful talk on deep federated learning in healthcare, with two recent novel works of FedDis to tackle the long-tail data distributions in different hospitals, and FedPerl to make better use of unlabeled data in semi-supervised federated learning. **Kaleem Siddiqi**,







McGill University, gave an interesting talk entitled "*Seeing Through the Heart: Fiber Geometry at the Nanoscale*" on cardiac fiber geometry analysis and moving to micro scale to study cell membranes and sarcomeres, powered by classical computer vision techniques without using deep learning. **Yuyin Zhou**, Stanford University, gave an excellent talk highlighting clinical needs and key challenges of AI for medical imaging, and proposed to diagnose like a doctor by taking advantage of structural prior, data prior, and medical knowledge in real-world clinical applications. Mathias Unberath, Johns Hopkins University, gave a fantastic talk on quantitative endoscopy with a series of cutting-edge algorithms for extremely dense point correspondence via self-supervised learning, dense 3D reconstruction in endoscopy video, and ultimately towards dense monocular SLAM.



Coming Down to Earth: Satellite-to-Street View Synthesis for Geo-Localization

Aysim Toker is a PhD student at the Munich School for Data Science. Her first supervisor is Professor Laura Leal-Taixé and her second supervisor is Professor Xiao Xiang Zhu from the German Aerospace Center and the Technical University of Munich.

Her paper explores a novel geo-localization method that may just revolutionize our everyday GPS.

Nowadays, huge databases of satellite images from every part of the world are widely available via services like **Google Maps**. However, the coverage of street-level imagery is more variable. In this paper, the team take street-view images from unknown locations and match them against a GPS-tagged satellite database to find the closest satellite image.

As humans, we are more used to seeing street-view images than we are images taken from above, so it may not be immediately obvious to us when comparing a satellite image to a street-view image whether they are showing the same location. This is made even more difficult because they are likely to be taken at different times of the day by different cameras. That is where computer vision and deep learning can help.

"*When we try to geo-localize a given street view, there is a huge domain gap between the satellite and street image,*" Aysim explains.

"*We propose to solve this by taking a satellite image, performing some simple mathematical transformations, and then synthesizing a content-preserving street-view image using generative adversarial models – basically, conditional GAN.*"

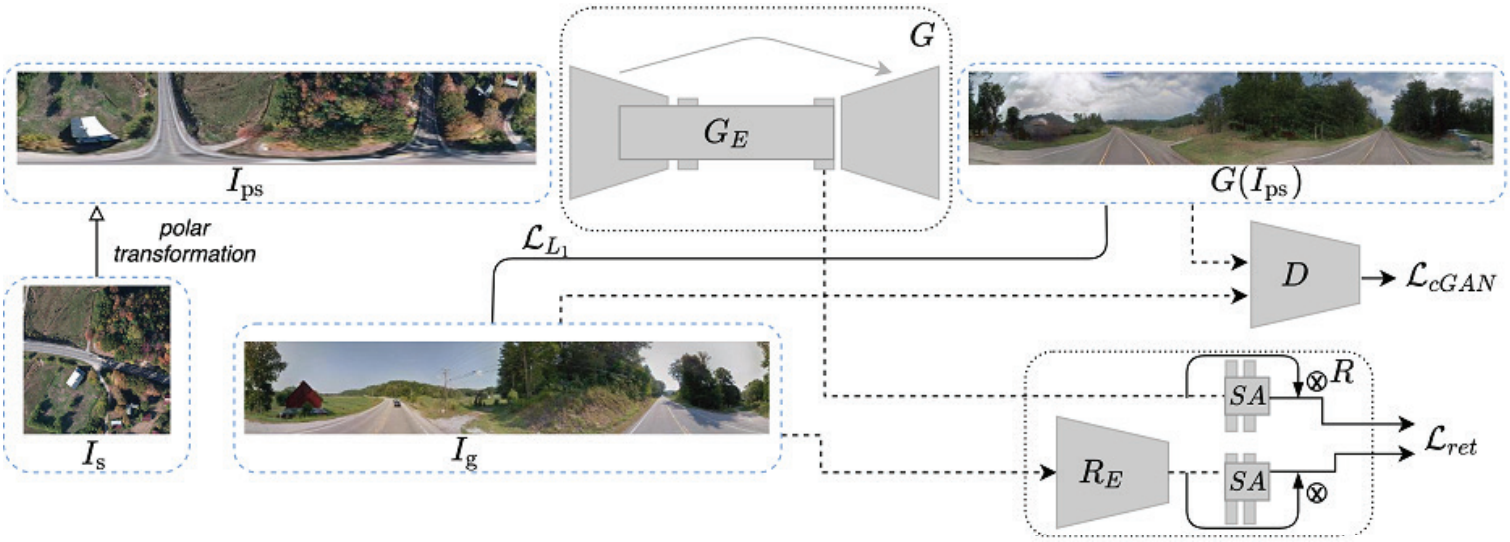
She says that **designing the architecture** was the really challenging part. They didn't know at first that when synthesizing a realistic image and geo-localizing the same image, the two aspects of the learning procedure interact and reinforce each other.

"*This mutual reinforcement was really exciting because it was done in one single architecture,*" Aysim tells us.

"*I think it will encourage other people to learn multiple things by fusing some tasks.*"

One major application of this work in the real world could be to improve the accuracy of GPS. As we all know very well, the GPS we use in our cars every day is good, but it is not always accurate enough. Thinking about immediate next steps, Aysim says they are currently working on a solution for finding the orientation of an image if the street view is not oriented.

We ask Aysim if anything ever went wrong – did the model ever find itself in completely the wrong place?



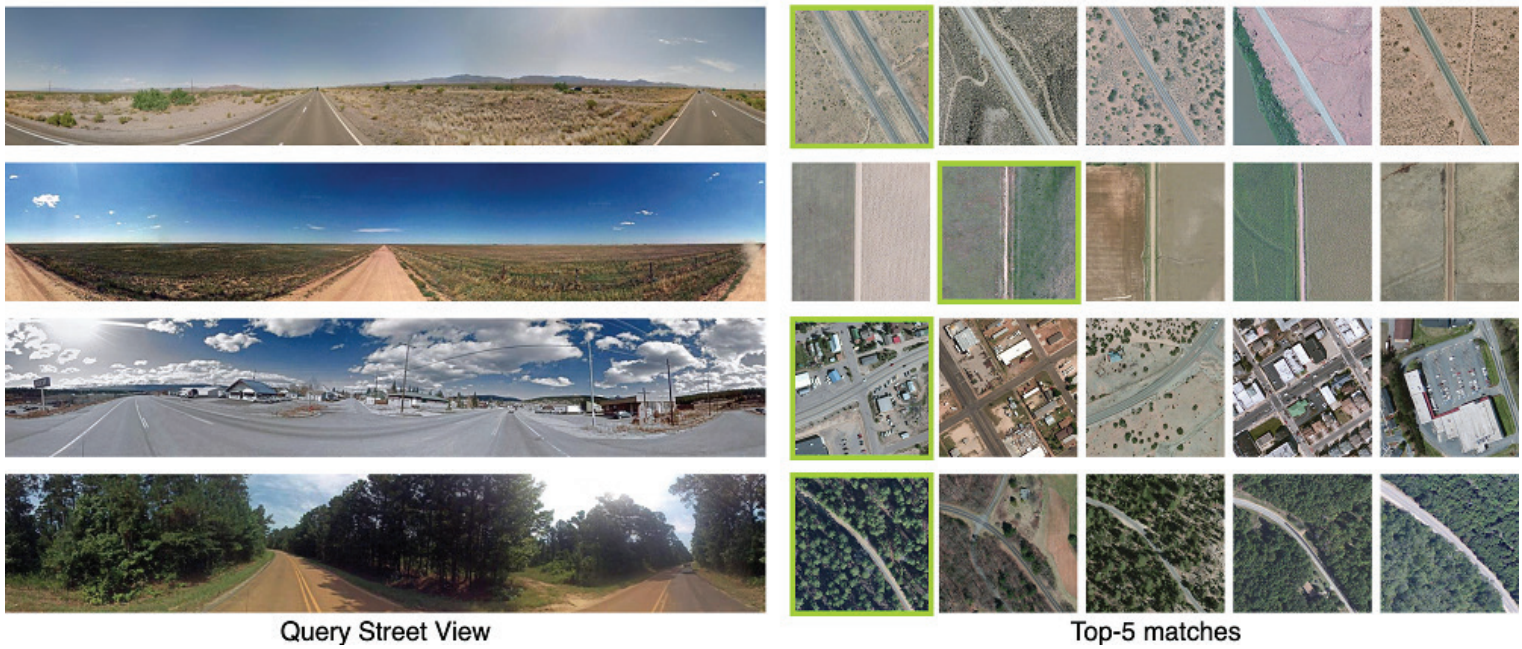
"*It never happened!*" she replies, laughing.

"*In the paper you will find some examples of street views with their correct matches ranked. They all look super similar. This is really smart actually!*"

Aysim, who is originally from İzmir in Turkey, has another two years to go with her PhD, and is still thinking about what she will do afterwards. She says the idea for this work came from her first supervisor, [Laura Leal-Taixé](#).

"*Laura is a really great supervisor,*" she smiles.

"*She's super helpful and creative. More a colleague than a supervisor I would say. It's really great to work with her!*"



And the winners of the 2021 Young Researcher Awards are...

Computer Vision News. Spotting talents in advance.

Subscribe now for free!

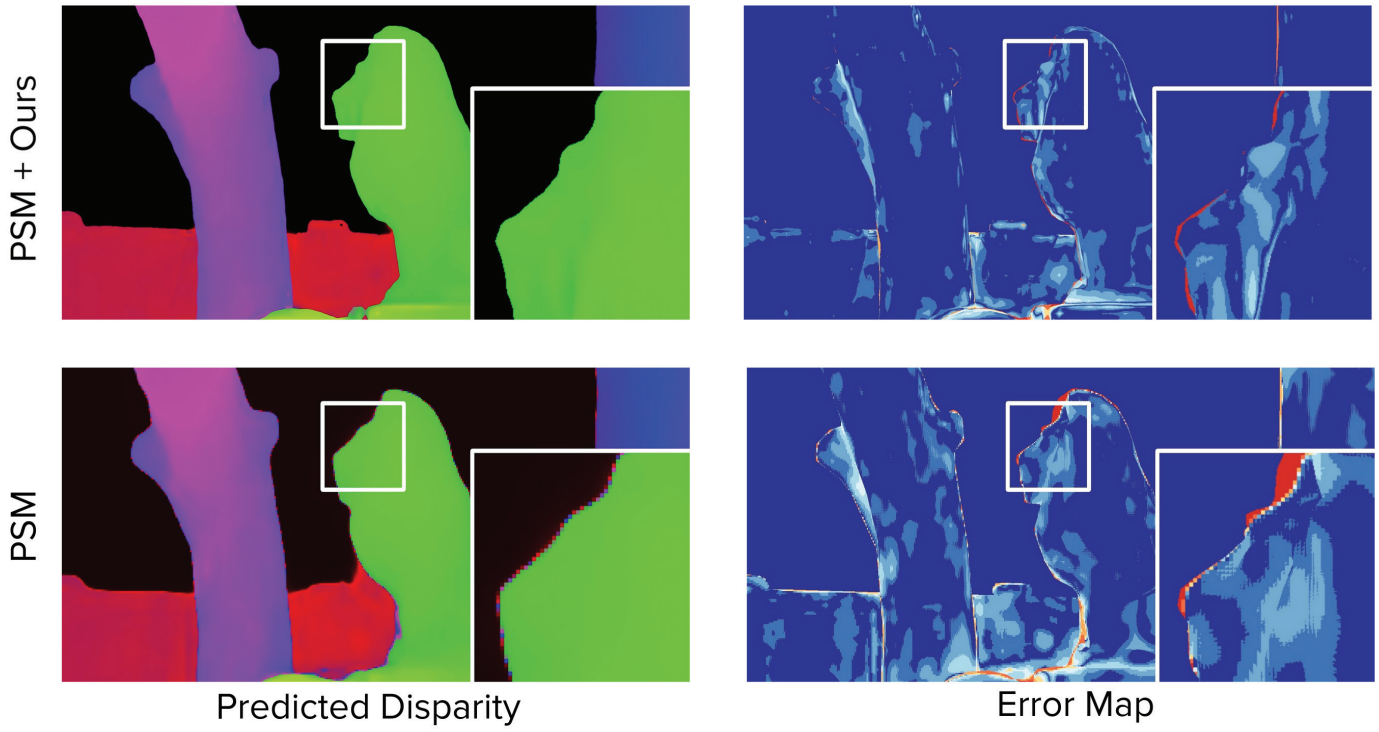
SMD-Nets: Stereo Mixture Density Networks

Fabio Tosi is a postdoc at the University of Bologna and a visiting PhD student at the University of Tübingen and Max Planck Institute for Intelligent Systems where he joined the Autonomous Vision Group.

Yiyi Liao is a postdoc in the Autonomous Vision Group where she and Fabio work under the supervision of Professor Andreas Geiger. Their paper focuses on stereo matching.

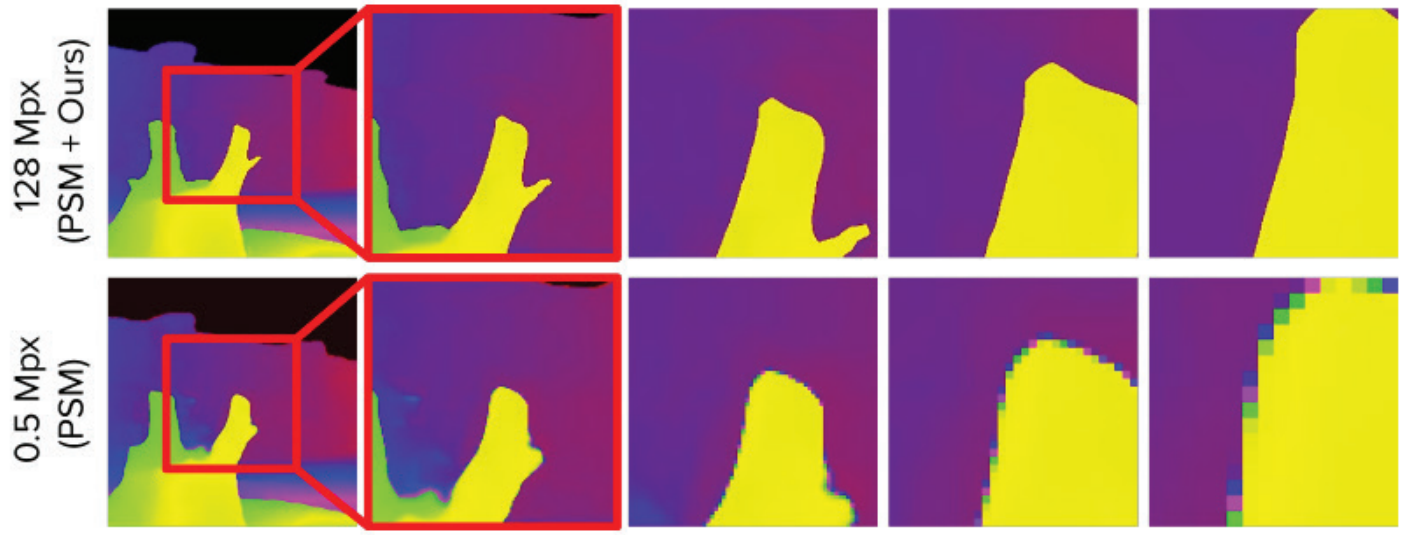
Stereo matching is a long-standing task in computer vision. It aims to recover the dense correspondences between image pairs to recover their geometry. Recently, with the rise of deep learning, convolutional neural network methods have replaced more traditional stereo matching methods. However, there are still two major problems that remain unsolved: predicting accurate depth boundaries and generating high-resolution outputs with limited memory and computation.

The overall goal for this work is to build a stereo matching algorithm that can work at a very high resolution and predict sharp and precise object boundaries. The team have a sensor in their lab that captures at 12Mpx resolution, which they want the algorithm to work with, and an algorithm capable of providing precise 3D geometry was desirable.



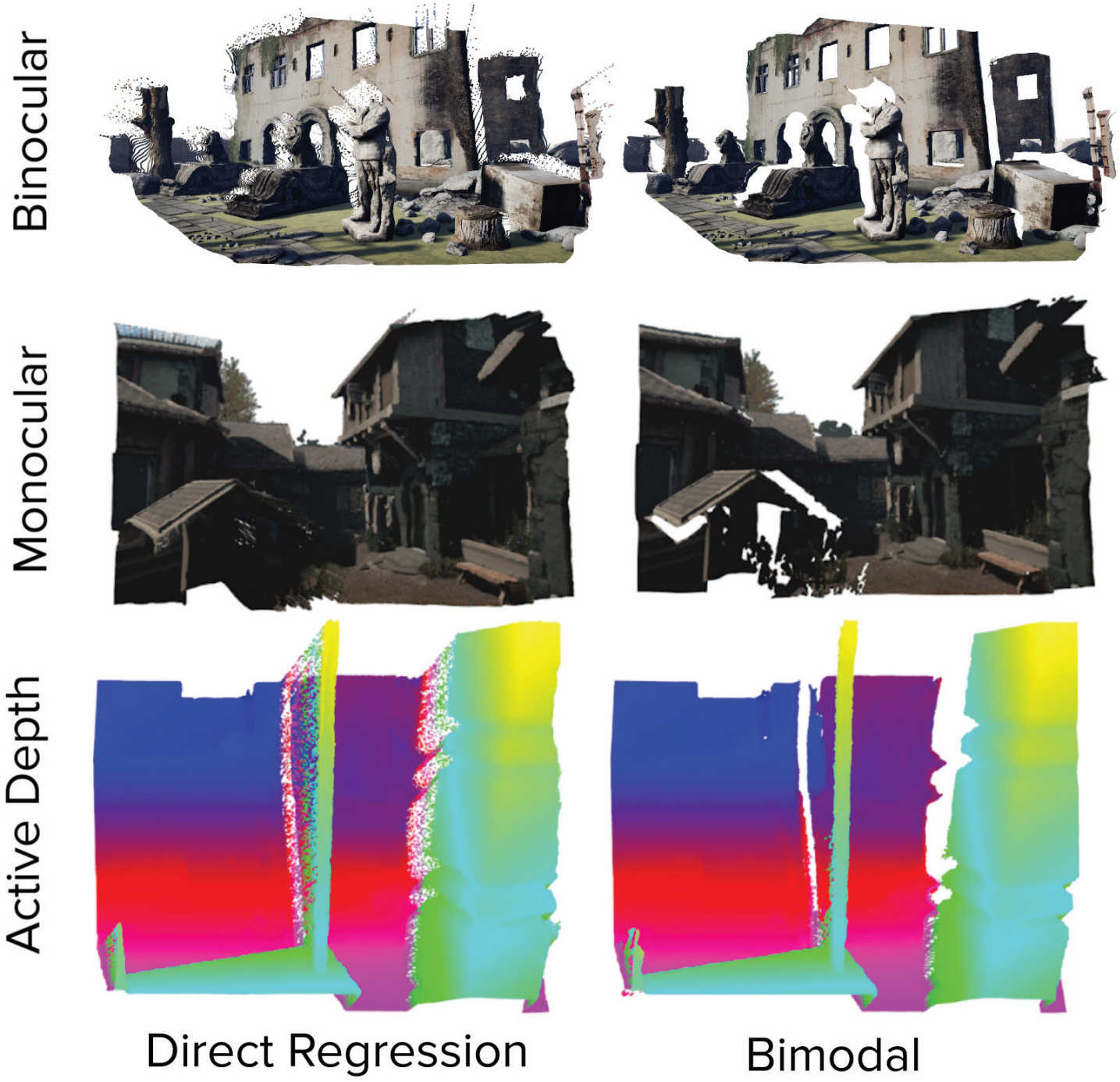
The overall goal for this work is to build a stereo matching algorithm that can work at a very high resolution and predict sharp and precise object boundaries. The team have a sensor in their lab that captures at 12Mpx resolution, which they want the algorithm to work with, and an algorithm capable of providing precise 3D geometry was desirable.

The team felt uncertainty was important. Typically, when considering uncertainty, a single-modal distribution is used. Fabio and Yiyi thought a multi-modal distribution, in particular a bimodal solution, would be a better alternative for stereo matching. This allows recovery of the sharp object boundaries. They demonstrated the flexibility of their technique by improving the performance of a variety of stereo backbones.



"I personally find the simplicity of this idea really exciting," Yiyi tells us.

"It complies with almost all different stereo backbones or even monocular depth estimation and active depth tasks. It's very general. It can be applied to many things.".



Fabio, who was already an expert on stereo matching when he joined the team, adds: "*I never imagined a general framework capable of predicting depth at arbitrary spatial resolution while keeping precise and sharp estimates at object boundaries. I'm still impressed by the elegant formulation and the results achieved.*"

In terms of next steps, with the output representation being so general, they plan to extend it to optical flow and self-supervised depth estimation. They are also interested in 3D reconstruction, which requires depth fusion of multiple images.

Thinking about challenges, the team faced one big one in particular, which was trying to work together during a pandemic! With Fabio in Bologna and other team members including Yiyi and **Carolin Schmitt** in Tübingen, they have conducted the entire project remotely and are yet to actually meet in person.

"*We organized it quite well even with this remote collaboration!*" Yiyi smiles.

"*We had great teamwork and worked together well to overcome any challenges. For example, at the beginning we had this output representation and we were thinking how we should train it. We came up with all different loss functions, like how to disentangle the foreground and background, which requires some extra supervision. In the end, what's surprising is that actually with a simple likelihood-based loss function then it works perfectly without extra supervision as we were proposing.*"

The initial idea for this work came from Professor Andreas Geiger. This is the second time we have spoken to somebody from his team, following our interview with Despoina Paschalidou at CVPR last year.

"*He is an excellent supervisor!*" Yiyi beams.

"*He gives very insightful feedback, both at the high-level and at a technical level. Whatever problem you are having, he is able to point you to the correct solution. I think he's really supportive – and I know everyone in our group agrees!*"

Fabio adds: "*Working with Andreas and his team was really exciting. The way they work and deal with problems is impressive. They opened my mind and taught me how to tackle research in a very different and effective way. I am extremely grateful to Andreas for this and for giving me the opportunity to join his incredible group.*"

BRepNet: A Topological Message Passing System for Solid Models

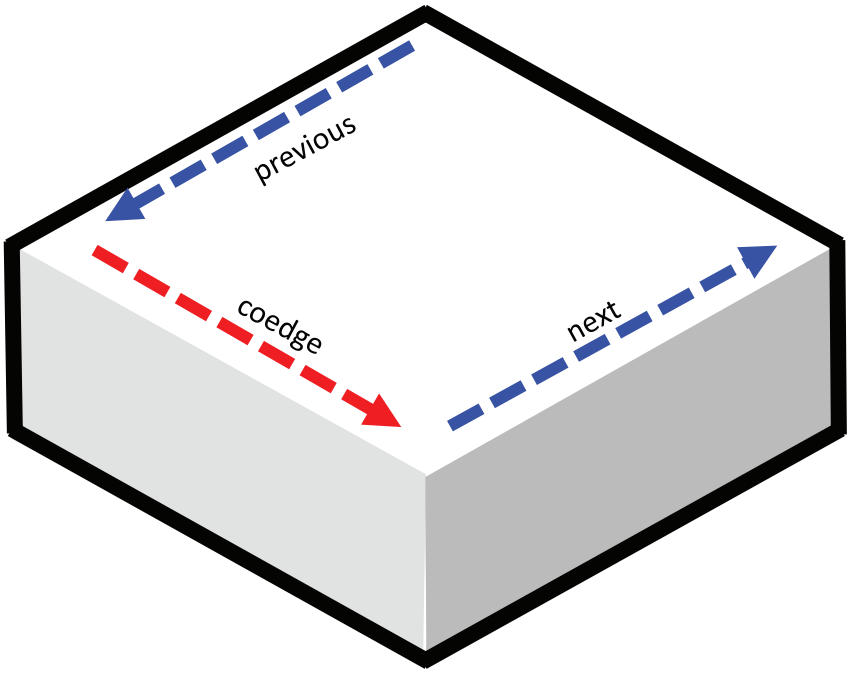
Joe Lambourne is a Senior AI Researcher at Autodesk Research with the Autodesk AI Lab. His paper proposes a neural network architecture designed to operate directly on B-rep data structures.

I begin by telling Joe what an honor it is to speak to him, as **Autodesk** played such a defining role at the start of my career in the 1980s. Back then, I was selling PC hardware and software in Paris, and **AutoCAD** was one of my products!

"That's wonderful. Well, obviously things have moved on a bit since then!" Joe laughs. *"Now, we have products like **Fusion** and **Inventor**, which are full **3D CAD modelers**. They all create models in the B-rep solid model format, which is the de facto standard for defining 3D geometry in industrial CAD."*

Until recently, machine learning has focused on representations like point clouds and triangle meshes, which lose a lot of information that is present in B-rep models. With the availability of open-source CAD modelling kernels like **Open Cascade**, which give people the ability to read B-rep models in STEP format, Joe and his team have picked the perfect time to exploit this gap.

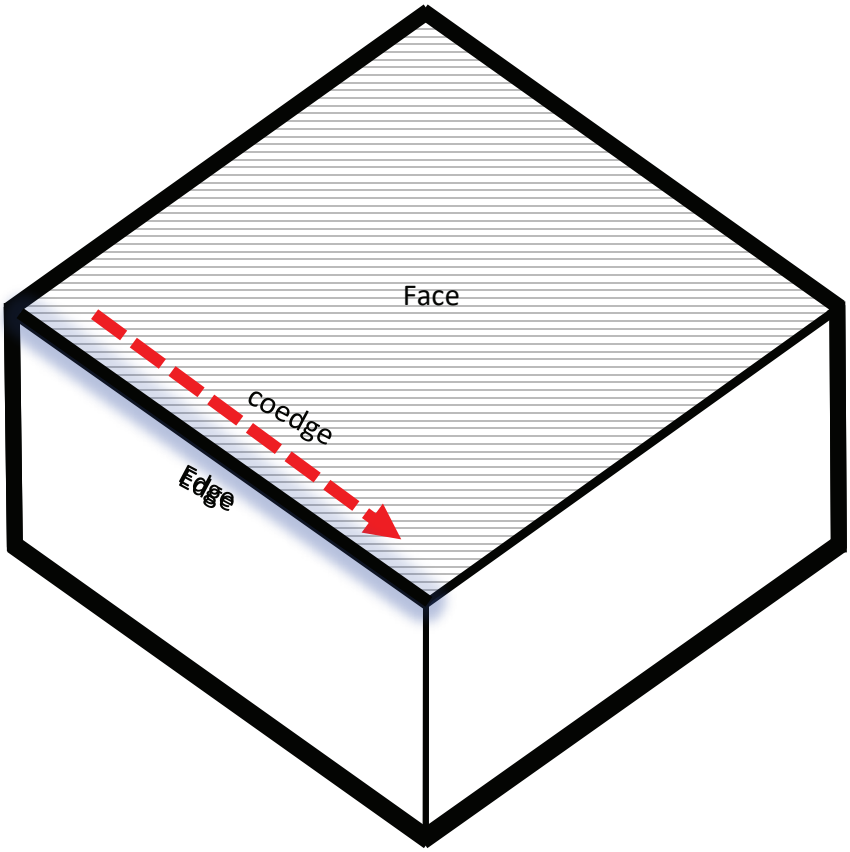
The **BRepNet** architecture is motivated by the realization that graph neural networks do not have any concept of ordering of one node around other nodes. They use symmetric aggregation functions to combine and aggregate the messages which are coming from neighboring nodes at each step. With the B-rep data structure and with solid models in general, the boundary representation is defining a manifold. It knows the ordering of any neighboring entity around a particular directed edge in the model.



The B-rep data structure is defined around oriented edges call coedges. In the data structure, some relationships are known about the neighboring coedges which are in the model.

For example, there are pointers pointing to the next coedge in the loop around the face and the previous coedge in the loop around the face. Also, there is the mating coedge on the adjacent face. This tells us which faces are next to which other faces.

There are also pointers to the parent face and the parent edge. With graph neural networks like the traditional message passing networks, the ordering of these different nodes around this red node in the middle is unclear, so you must use symmetric aggregation functions in order to combine

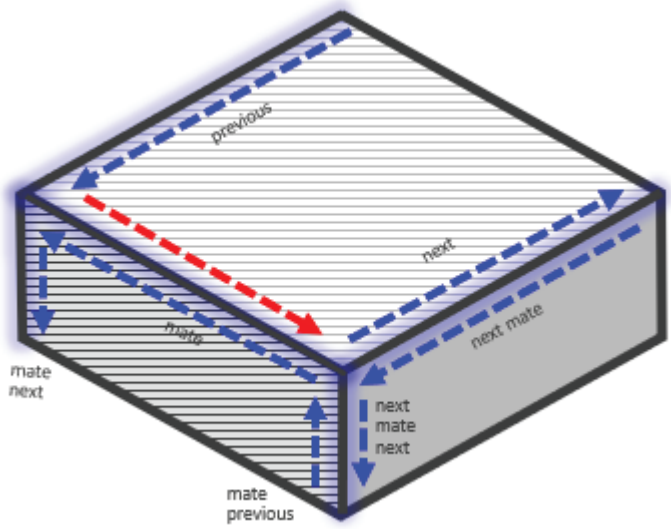


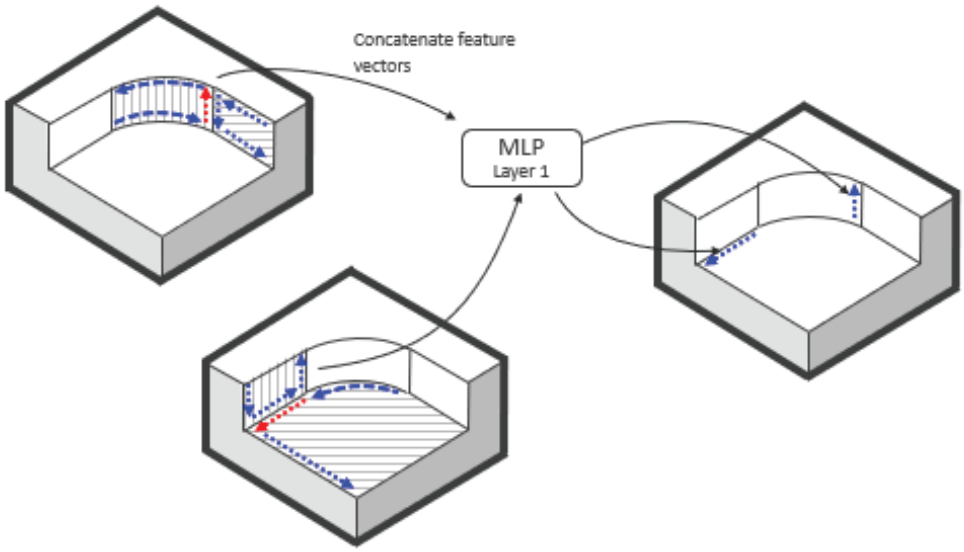
those messages in the convolutional scheme.

"*With the B-rep structure, we actually have a way to do better than that,*" Joe explains.

"*We can describe any neighboring edge, face or coedge using this idea of what we topological walks. A topological walk is a list of instructions which tell us how to move from one coedge to another. If we think of the instruction list to go to the next coedge and then to its mating coedge and then to jump to the parent face, then using this sequence of instructions, we have a way to address this face from this coedge in a completely unique way. This sort of thing is just not possible with a graph network, where if I want to talk about this node in the graph, all I can do is I can talk about the neighboring nodes. I can't choose a specific one.*"

The BRepNet architecture is designed specifically to take advantage of the additional capability stored in the B-rep models. Using this set of topological walks, you can describe neighboring faces and all neighboring edges and then take the data of whatever feature vectors you choose





to use to describe the initial geometry there. They can then be passed through a multi-level perceptron (MLP), which is like a learning set of weights, to start to recognize patterns in these local structures. Then the learned representation is attached to this representation of the same coedge but in the next layer of the network. This can be done for every coedge in the model and then max pooling used to generate new embeddings for the faces, which can be passed on to build a multi-layer architecture. This is the core idea of BRepNet.

The team have another accepted paper at CVPR this year, called **UV-Net: Learning From Boundary Representations**. This proposes a way to provide more geometric information about B-rep models to neural networks by describing the actual surface geometry using regular grids of points which are regularly spaced in the UV parametrization of the surfaces.

Currently, BRepNet uses very simple input features to describe the geometry and uses the Fusion 360 Gallery Segmentation Dataset to help test the power of these algorithms using real-world rather than synthetic data. Moving forward, they want to combine the point grid concept in the **UV-Net architecture**, which is a good way of describing the geometry of the faces, with the BRepNet convolution scheme, which is an efficient way of doing convolutions on the boundary representation. This will provide a better way to understand the models and help users of AutoCAD software to automate many of the tedious selections that they are making now.

"Obviously with BRepNet we have a very flexible way of defining which entities are used in each convolution," Joe tells us.

"When we did the experiment, the convolution kernel which gave the best performance turned out to include exactly the entities which were present in the winged edge data structure invented by Baumgart back in 1972. This shows how classic geometric modelling ideas can play a part in modern machine learning!"

"This shows how classic geometric modeling ideas can play a part in modern machine learning!"

This area of technology is growing in popularity, with new publications showing the possibility to build B-rep models using sequential data, and transformer models.

*"The **DeepCAD** paper published recently on arXiv has everyone very excited because they see the potential to not only automate things like manufacturing, but to be able to create new designs as B-rep models as well,"* Joe says.

"It's a really exciting area for people wanting to explore something new in machine learning to get involved with."

If you are interested in getting started working with solid models, you will find the public code for BRepNet here.

Ann Kennedy is an Assistant Professor of Physiology at the Northwestern University School of Medicine. She has opened her laboratory for the Theoretical Neuroscience of Behavior in the Department of Physiology at Northwestern University.

More than 100 inspiring interviews with successful Women in Computer Vision await for you in our archive!

Ann, you have recently opened your own lab.

Yes, we started this past October.

Best of luck! Also, congratulations because you're in the team that won Best Student Paper at CVPR 2021.

Yes! Jennifer Sun's paper won best student paper. Jen is really the one who deserves the congratulations here. We've been working together on adapting computer vision tools to problems in neuroscience. She's been really fantastic at getting to know the problems that the field is trying to solve.

What do you work on in neuroscience?

My lab is a computational and theoretical neuroscience lab. We collaborate with experimental groups

in neuroscience, with the goal of better understanding the structure of animal behavior and how it's controlled by the brain. Within the lab, we work with computer vision tools for doing automated pose estimation and behavior classification, mostly in mice. We also work with groups that are recording the activity of neurons in different parts of the brain. We develop methods to relate what those neurons are doing to the animal's behavior to try to understand how the brain is encoding behavior and, in the big picture, how circuits of multiple interacting parts of the brain giving rise to behavioral choices in animals.

Everything you said so far raises so many questions! Since you are studying how the cognitive process works, do you think our brain is functioning well? Also, since you are studying animals, do you think the brain is functioning as it should? Would you suggest any improvements?

[*laughs*] That's a good question! During my postdoc, we were looking at very evolutionarily conserved circuits in the brain, or governing survival behaviors, like defending your territory, escaping from predators, feeding, those kinds of behaviors. These are parts of the brain that have been around for a very long time. They have been genetically wired to keep animals alive - to survive and reproduce. These areas have been under tremendous evolutionary pressure. It's incredible how much you can accomplish just by wiring these things up genetically without any sort of supervised training of the circuits. Animals are born and sometimes run within a couple hours of being born; they can find food, evade predators. All of that is encoded in their genes and to some extent there in the brain from birth. Obviously, there is learning and experience on top of that, butthe level of things that the system can achieve without any supervised learning or reinforcement learning is really fascinating to me. That is something brains do very well!

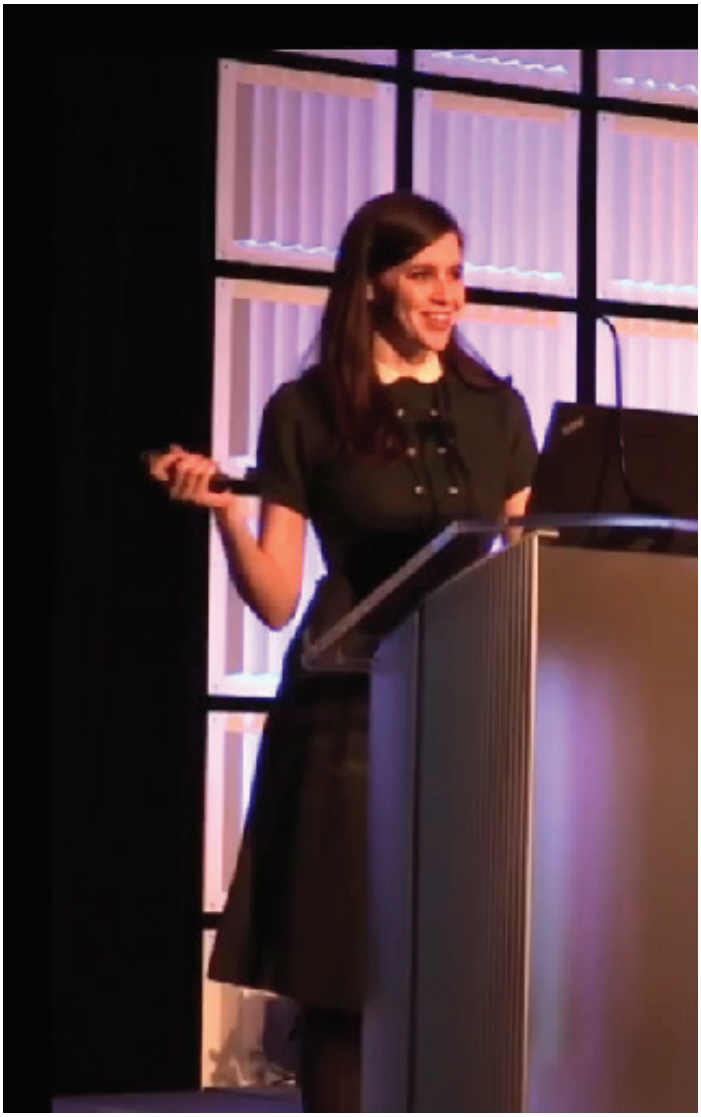
Regarding our learning process, I have noticed that if you show me several fire extinguishers, I will quickly be able to recognize one. Why does a machine need to see the same image thousands of times before recognizing it? How does the brain do this more effectively?

I guess you're talking about few shot learning, the ability to recognize an object the first time you see it. That's something that people are working on in object recognition: transfer learning or meta learning or few shot learning. This isn't my area of research so I probably won't do a great job describing this, but if you learn to recognize enough objects in the world, like cats and boats and airplanes and trees, then you've built a repertoire of features general enough that you can train a new part of

your brain or a new neural network to recognize a fire extinguisher with a very few number of examples.

Do you ever feel uncomfortable working with mice?
I don't personally work with mice - mostly I collaborate with experimental labs that collect this data. Although I have a small component of my lab that is not up and running yet but will be collecting behavioral datasets in mice.

So somebody else is doing the dirty work?
[*laughs*] I guess so, yeah. It's a part of the job!



It doesn't seem like the most attractive part of the job anyway. Do we really need to work with animals in order to understand ourselves better?
That's definitely a thought that has arisen in theoretical neuroscience since the emergence of deep learning - there are people who are studying artificial neural networks as opposed to working with animal data. I think it really depends on the question that you're trying to ask. If you care about ways that any neural system could learn, you could do that in an artificial neural network. If you're looking at how things are specifically happening in the human brain and animal brains, and how we treat diseases and disorders, then we still need some access to the biology. We need to see the real side of things. It really just depends on your priorities.

I guess the priority is studying mammals. Is that right?
There's a vibrant invertebrate neuroscience community as well. I collaborate with a postdoc who studies the nervous system of jellyfish. My postdoctoral lab has a cohort of people who work on the fly nervous system. I think there's a lot you can learn from that too just because insect nervous systems are so different from mammalian nervous systems. In mammals, a given task will be associated with a pool of thousands or tens of thousands of neurons

collectively computing things; whereas in flies, you might just have one neuron in the entire brain that is driving a certain behavior. You can genetically activate just that neuron and study it. It's really a different way of building a brain.

So if I say to somebody, you have the brain of a fly, that's a serious, serious offense.
[*laughs*] Sure!

It means the person has only one neuron!
One neuron for a given behavior. They can still do a lot with that.

When observing the mice, did you ever see anything really adorable?
You definitely learn a lot just from looking at the videos of the raw behavior. The work I'm doing to automate behavior detection is all based on supervised classification of behaviors we've defined after years of investigation. There's a lot more that animals are doing that we're not necessarily looking for. I don't know if I have any great examples in mice. They are very social. There are different strains of mice that have different personalities: some are more aggressive; some are a lot more relaxed. It's interesting realizing that these genetic backgrounds of animals have so much influence on their personalities. But even within a strain of mice that are all genetically identical to each other, you'll have mice that are more aggressive and mice that are more submissive. The interaction between nature and nurture, genes and experiences, is really fascinating. In general, a lot of people that work on behavioral neuroscience get into it because they see animals behaving in the world, and they want to understand how that decision making process works. What makes animals act in a certain way?

What gives them particular instincts?

"What makes animals act in a certain way? What gives them particular instincts?"

One last question about your work: of all the differences about humans and animals, which one is the most striking?

I like to think there are some differences between humans and mice! We have a much more developed cortex. We have more executive control than these animals do. Mice are prey animals, and they are very skittish. It takes a long time to teach a mouse to do something, for certain kinds of tasks. That's one of the frustrating things about mouse research. Say you want to study how an animal makes a decision, how it integrates sensory evidence and then make some choice about that evidence. It can take days to get a mouse to understand the kind of task you're trying to get it to do, and you have to just hope that it's doing the task the way you think it is.

I could talk to you for hours about your work, but I also want to ask a bit about you. How did a young woman like yourself end up here?

My parents were both computer programmers, so I learned how to code from my mom when I was pretty young. I was also interested in biology, so I split the difference by studying biomedical engineering in undergrad, learning a bit about biological systems but also learning math and learning how to code. That balance was important to me. My undergrad institution had a couple people who were doing modeling of the nervous system. That was just fascinating to me. It wasn't something that I knew was really an option until my senior year of undergrad, so I applied to a bunch of neuroscience PhD programs, not knowing what neuroscience research was at the time. I was lucky enough that I was accepted into one.

What was your biggest "wow moment"?

This specifically, what I'm doing now, it's hard to say. When I was a kid I was always super into robotics; realizing how hard it was to make robots do things was eye-opening to me. These things that seemed trivial to me, like recognizing an object or walking without falling over, that you don't even think about, turned out to be really hard to do when you sit down and try to reproduce those behaviors in a program or a machine. That was always fascinating to me, this distinction between what biology can do and what humans can build machines to do. Then a more concrete example is when I was in undergrad. I had a class where we did a problem set on training a perceptron to discriminate pictures

of animals versus pictures of non-animals, and it worked! It was kind of just magical! You gave it all of these pictures, and suddenly, it was able to distinguish these two things. That was just wild to me, that you could build something like this to solve a task and not really understand how it was solving it. It would just magically work.

Does being a researcher in your field make you believe more in magic?

Oh no! It's not magic. It's just simple learning rules implemented in a big network of neurons.

Do you notice any common mistakes that junior researchers make?

Attending talks and conferences and getting out there is something that made a big difference to me early on in grad school. It's common to have the feeling that "I shouldn't go to this meeting until I really understand what people are working on", until I know the field. But you go to these meetings because you don't know the field! Attending conferences and asking stupid questions is how you get to know an area of research. You shouldn't hold

yourself back from doing these things just because you feel like you don't know what you're doing. That is definitely something that I think is important early on in your research career. Also, be open to things. Don't just go to the talks that directly relate to your research. Go to a lot of talks because you never know, 10 years down the line that thing that you learned about, that you didn't really care about, may resurface.

see not just what one little piece of the brain is doing, but how that piece is interacting with all of the other pieces. We've gotten there with some very small organisms, like C. elegans and larval zebrafish - we can see all of the pieces interacting like you can do with an artificial neural network. larval zebrafish - we can see all of the pieces interacting like you can do with an artificial neural network.

Let's get back to magic. If you had a magic wand, what is one thing that you'd like your lab to discover?
That's tough! Right now, we mostly want more people.

And if you could fulfill one dream in your career, what would it be?
These are big questions! One thing that people dream of in neuroscience is recording all the neurons in the brain of behaving animals, so you can

So that would be a dream, just to see all of the pieces of the brain in motion at once.

I hope you will be the one to jump the hurdle!
I hope someone does!

More than 100 inspiring interviews with successful Women in Computer Vision await for you in our archive!

COMPUTER VISION EVENTS

MIDL
fully virtual
July 7-9

Mass Data Analysis...
New York, NY
July 11-13

MIUA
Virtual
July 12-14

SPIE Optics + Photonics 2021
S.Diego, CA
(options for remote)
Aug 1-5

Ai4 2021
fully online
Aug 17-19

Medical Augmented...
Online
30 Aug - 10 Sep

MedTech Conference
DC - Minneapolis - Virtual
27-30 Sep

MICCAI 2021
Virtual
27 Sep - Oct 1

Meet us there

ICCV 2021
Virtual
11-17 Oct

Meet us there

IMVC
Tel Aviv, Israel
Oct 26

Due to the pandemic situation, most shows are considering to go virtual or to be held at another date. Please check the latest information on their website before making any plans!

46 Congrats, Doctor!

Jie Ying Wu recently completed her PhD at the Johns Hopkins University. The goal of her research is to enable surgical robotic systems to better understand surgeries. This constitutes a step to create intelligent assistance to surgeons during operations. She will start as an assistant professor at Vanderbilt University in January 2022. Congrats, Jie Ying!!!

Jie Ying separates a robot-assisted surgery (RAS) into 3 parts: the robot, the surgeon, and the patient scene. For each of the parts, she learns representations of the data through cross-modal self-supervised learning. Cross-modal self-supervised learning exploits the synchronicity of the signals inherent in the system and learns to map the signals from different modalities onto each other. This could be used to reveal underlying patterns, such as surgeon gestures or skill level such as in [Wu et al, IJCARS 2021], or improve models of patient anatomy.

For modelling the patient scene, while pre-operative scans such as CTs capture the anatomy, they do not necessarily capture how organs would deform. Current methods for simulating soft-tissue deformation rely on accurate material parameters and boundary conditions, which are often unavailable in patients. We developed a machine learning method to correct the imperfect simulation based on observations. It is expensive to label data for learning deformations though since it is non-categorical: imagine the difference between labelling that this image contains a liver vs. labelling how the liver should deform if a robot interacts with it a certain way. Instead, we use observations of the scene as labels. In a very simplistic set up, we probe a soft-tissue phantom, shown in Figure 1, using a da Vinci robot and learn to correct simulations of how it deforms.

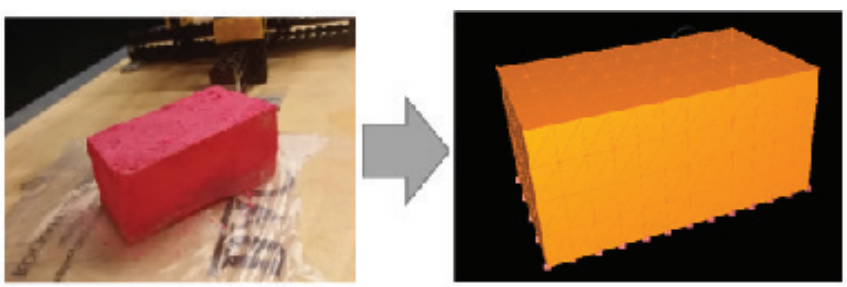


Figure 1: Soft-tissue gel phantom (left) and a model of it for simulation (right) [Wu et al., IJCARS 2020].

We build a model of the phantom in simulation. We can then use finite-element method (FEM) to model how the phantom should deform using material parameters we had measured. Since the robot tracks its instrument

movements, we can replay the interaction in simulation and see how well it matches real observations of the scene. To observe the interaction with the phantom, we mounted a depth camera and recorded point clouds of scene, such as the one shown in Figure 2. We trained a network to correct for the difference between the point cloud and the model. In a real surgery, the base simulation parameters could come from atlases while we learn patient-specific characteristics from observation.

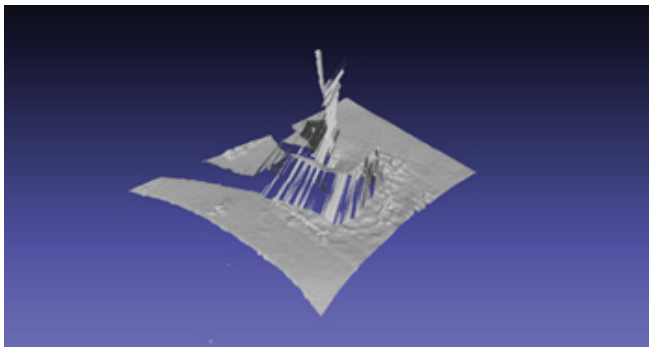


Figure 2: Point cloud observing interactions between the robot and the phantom.

In order to use these simulations for intraoperative guidance, we need to ensure the simulations are not only accurate but can run at real time speeds. While FEMs are the gold standard for accuracy, they are generally slow. Since we already have a neural network predicting corrections, we hypothesized that the network can also learn to predict the deformation step. We train graph neural network to learn to mimic the predictions from an FEM. This required the neural network to learn the dynamics of the scene and use it to predict successive deformations.

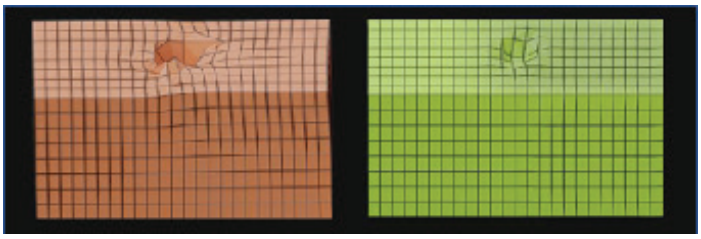


Figure 3: Deformation as estimated by the FEM (left) vs. the network (right). For the same sequence, the FEM took hours to simulate while the network simulated the deformation in real time.

We can see there is some accuracy trade off as the network's predictions are less deep and smoother than the FEM's. Nevertheless, the speedup realized by the graph neural network is considerable and we consider that it is worth pursuing a network-based simulator further for refinement of the predictions. Future goals for this project include learning to model more complex phantoms, such as the hysterectomy phantom in Figure 4, and using augmented reality to provide guidance to users through the real-time simulations.

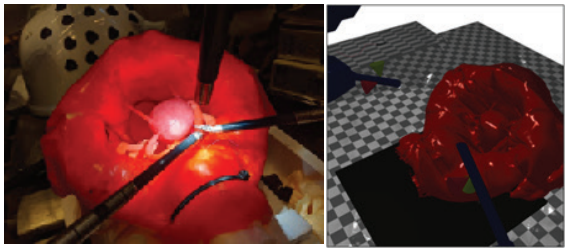
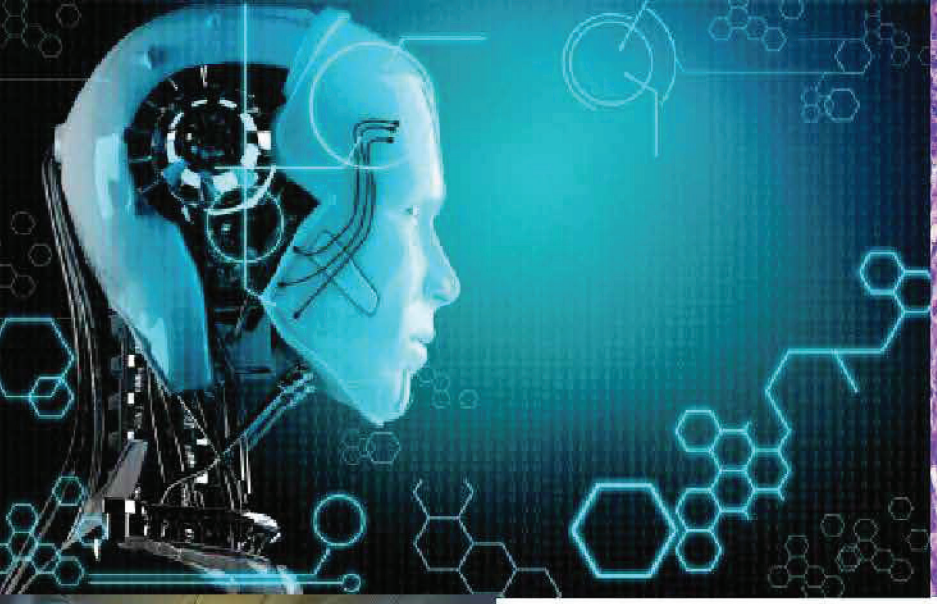
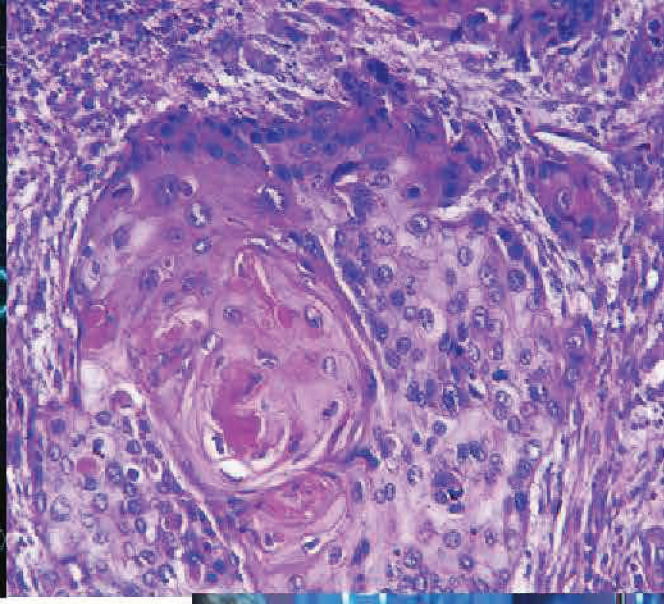


Figure 4: Hysterectomy phantom in the workspace of a da Vinci Research Kit (left) and a model of the same scene (right). The simulation scene is available here.

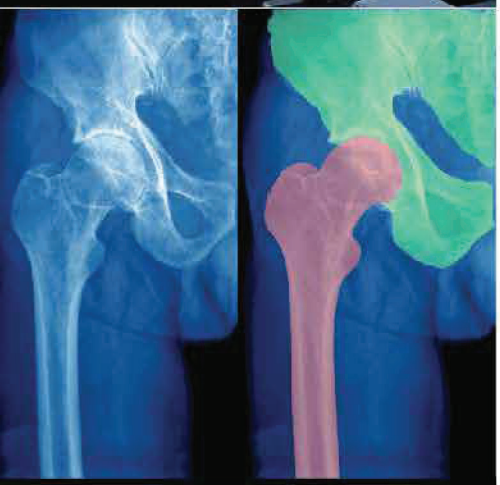
Jie Ying would like to thank her wonderful advisors Peter Kazanzides and Mathias Unberath for being ever supportive and encouraging. She would also like to acknowledge Adnan Munawar for contributing his simulators expertise for this work.





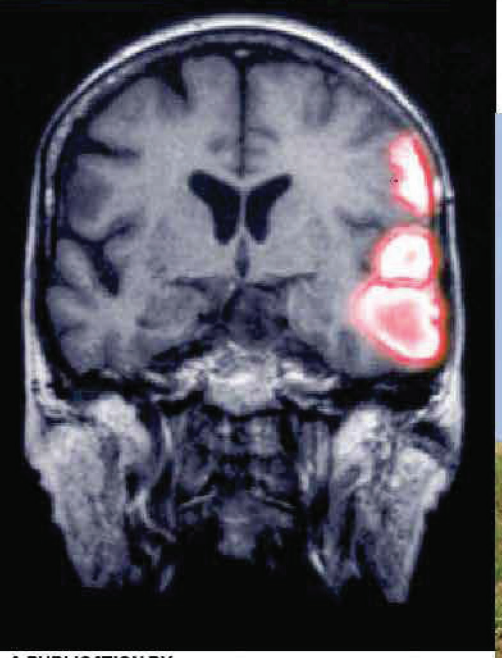
IMPROVE YOUR VISION WITH

Computer Vision News



SUBSCRIBE
CLICK HERE, IT'S FREE

The magazine of the algorithm community



A PUBLICATION BY

RSIP VISION

Global Leader in Computer