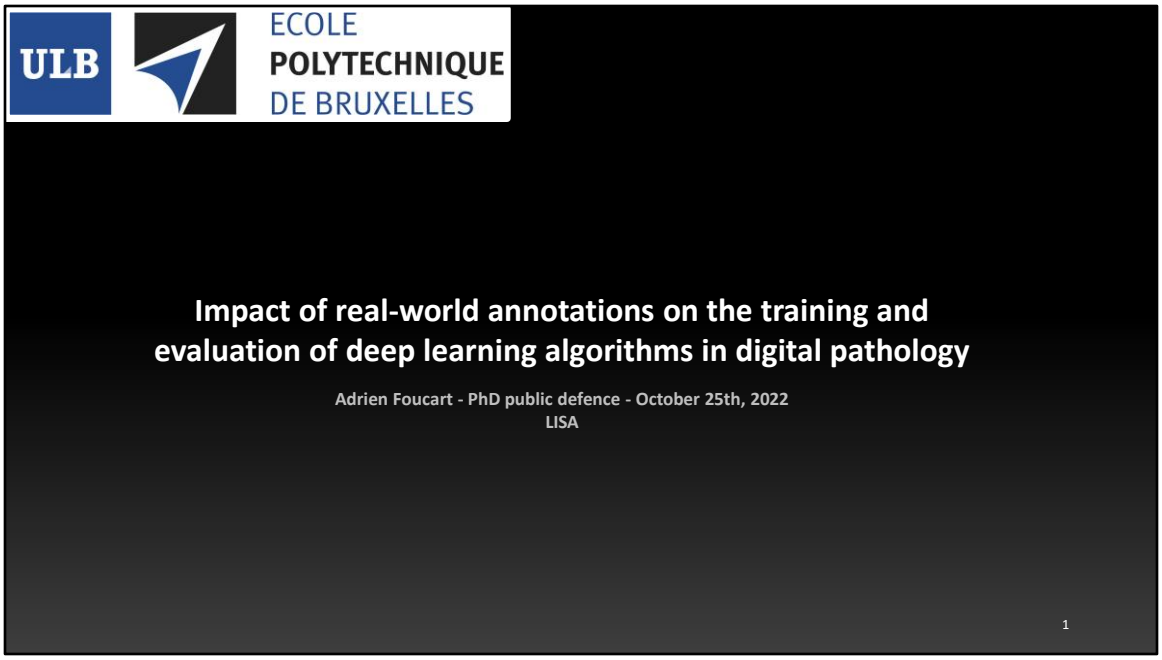


Good afternoon,
Thank you all for coming to this presentation, and to hear me talk about what I've been doing these past seven years.
Before we really get into the topic of the thesis, and this long title, I would like to give a bit of context. So let's start with the last words: digital pathology.

Impact of real-world annotations on the training and evaluation of deep learning algorithms in **digital pathology**

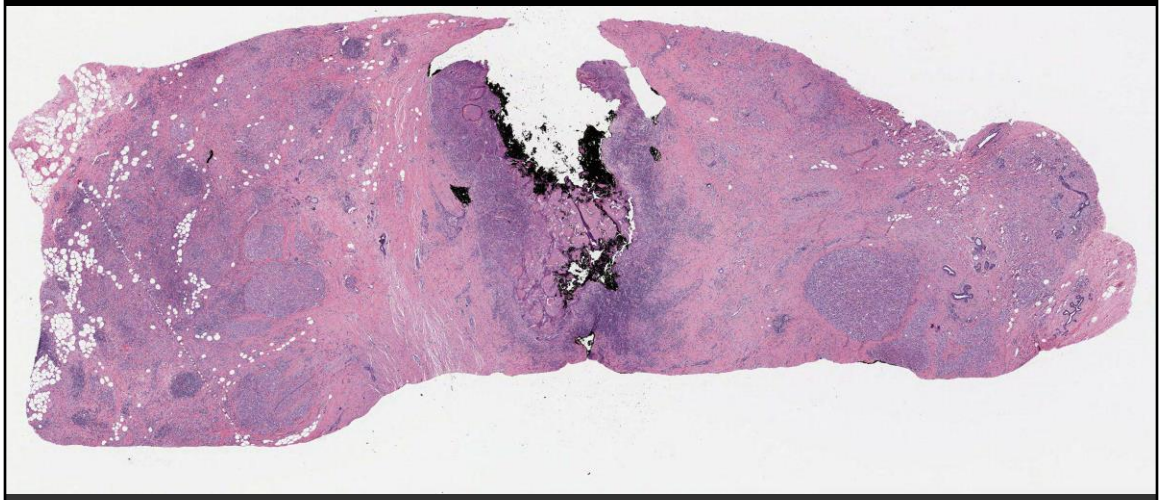
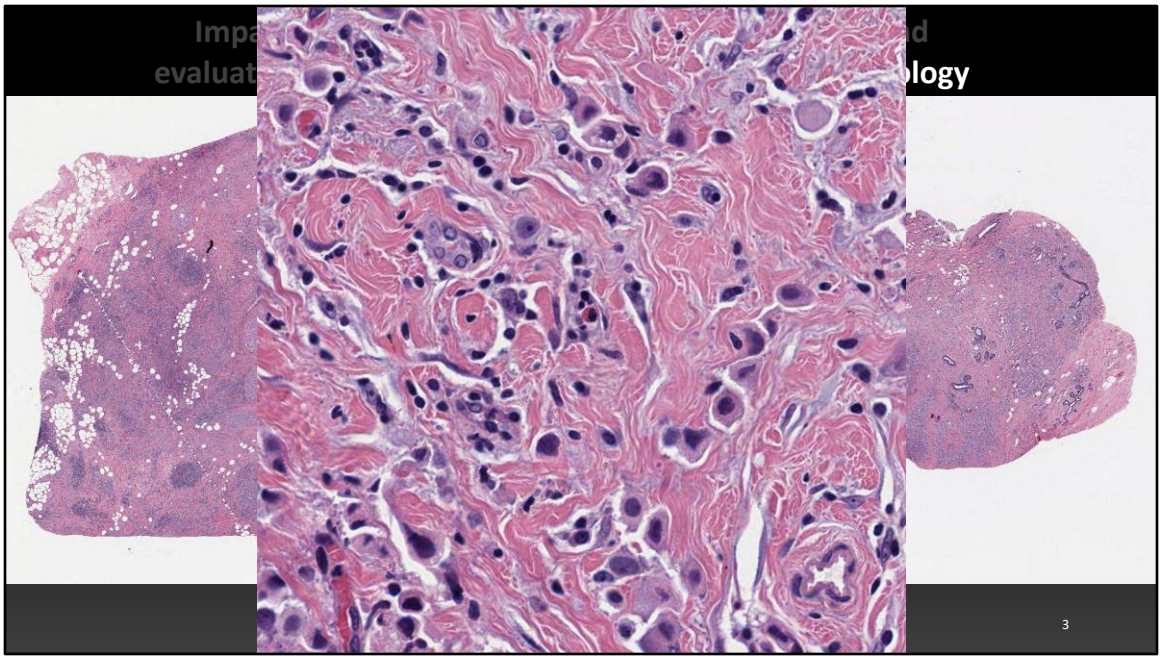


Image: The Cancer Genome Atlas.

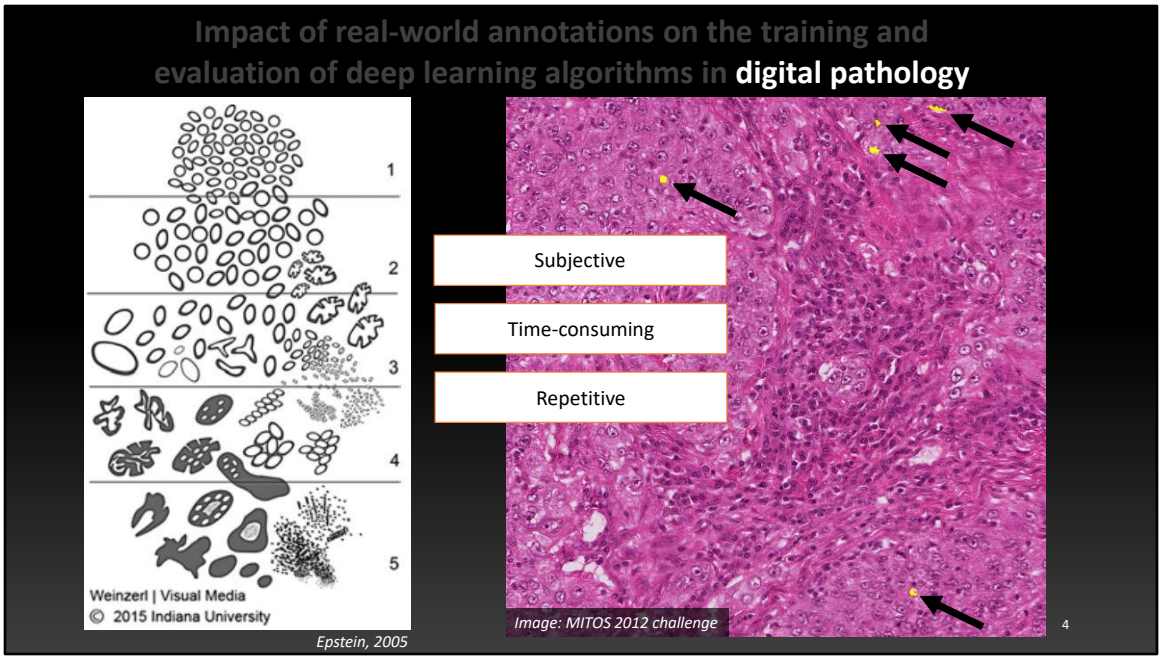
2

Pathology is the study of diseases. Let's take an example. After a mammogram, or a prostate exam, someone has a suspected tumor. To determine the severity of the tumor, and what to do about it, one very common step is to take a tissue sample, for instance with a biopsy. This tissue sample, after some fairly complicated processing, ends up as tissue slides which can be viewed under a microscope. Digital pathology just means that, instead of looking through the microscope, the image is scanned at a very high resolution.



These images are typically huge: from this relatively low magnification view, we can zoom in all the way to the level of the cell nuclei.

The digital copy can be viewed on a computer, stored in hospital archives, or potentially transmitted to other pathologists if a second opinion is needed.

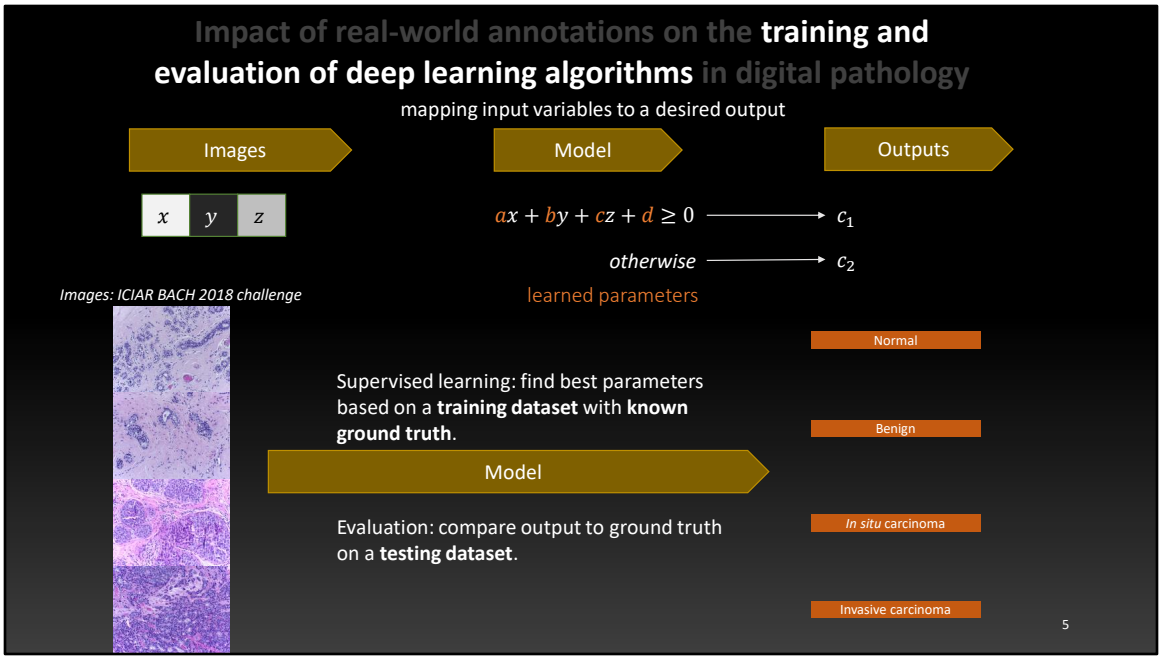


What are pathologists looking for in those images? There are lots of indicators that are used to grade tumors, but some common one are based on the shape of some cellular structures, like here with the Gleason grading scale based on the shape of glands in the prostate. Other are based on counting some particular objects, like lymphocytes or, in this case, mitotic cells.

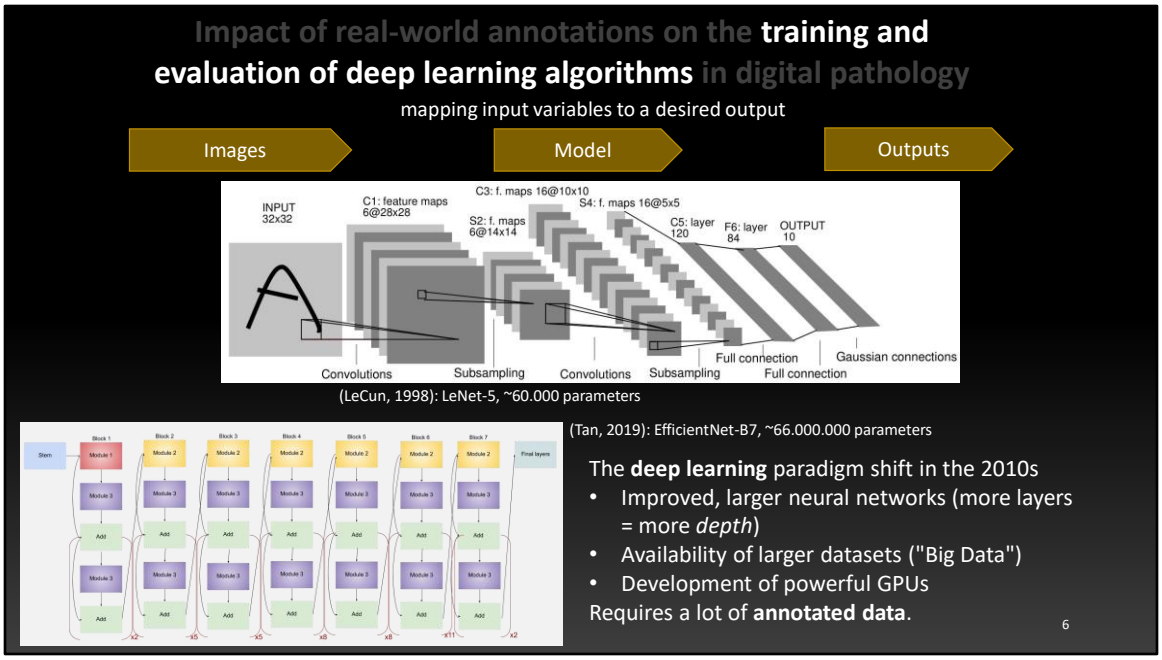
The problems are so complex that the criteria used by the pathologists are often very difficult to assess objectively, and there can be a lot of disagreement between the opinions of the pathologists.

Some of the tasks, like counting the mitosis, can also be very time consuming and repetitive. So it would be very nice to have some automated image analysis methods to help.

In the past decade, the most successful image analysis methods have been “deep learning” algorithms, so let me briefly explain what these are about.



Let's say that we have as input an image. To make things simple, we will say it has only three pixels. Let's call their values x, y, z. At the output, we have two classes: c_1, c_2. Between the two, we have a **model**. A model is a mathematical mapping between our input and our output. One of the simplest model we can have is a **linear** model. We could for instance have: output is c_1 if ax+by+cz+d >= 0, and c_2 otherwise. In this case, a,b,c,d are the **parameters** of the model. **Training the model** means finding the best values for these parameters, so that the output is as correct as possible. In supervised machine learning, "finding the best parameters" is done by using a **training dataset**, containing examples where the expected output is known.



Obviously, that model is a bit too simple. So what we can do is make it bigger, more complex… Neural networks do that: they combine relatively simple equations, chained together, to produce a model that is more complex, and has more parameters.

The network is organized in layers, built on top of each other. The first layers, closer to the input, learn to recognize small scale patterns and colors. Then, the following layers combine those patterns into more abstract concepts, shapes, object parts, and in the end these high-level, abstract features are combined to produce the desired output.

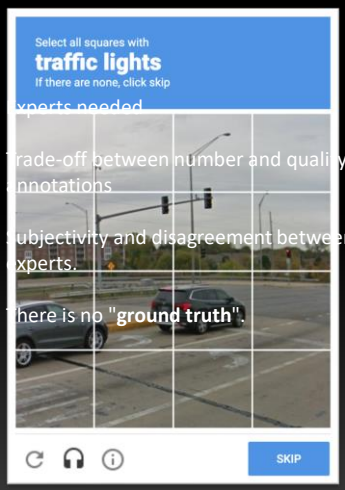
They will obviously also take larger images as input. In the 1990s, the best networks for image analysis had more than 50 thousand parameters. At the time, it was very difficult to train anything bigger than that, and the performances where relatively limited.

In the 2000s and 2010s, however, there was a big shift in image analysis. As it became much cheaper to store and transmit data, larger datasets became available. And with the rapid development of GPUs, the amount of processing power available to train the networks increased dramatically. Modern deep networks have millions or tens of millions of parameters, and can model very complex functions.

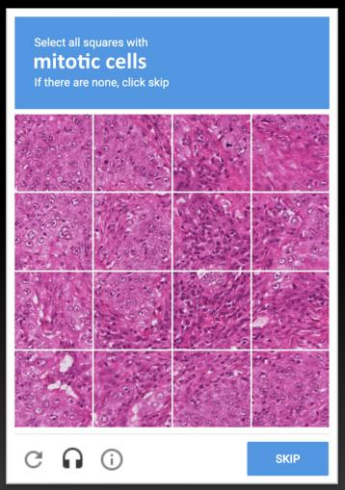
But to do that, they also need a lot of data. And, ideally, a lot of **annotated** data,

meaning that the expected **output** is known for every example given to the network.
So how do we get annotations?

Impact of **real-world annotations** on the training and evaluation of deep learning algorithms in digital pathology



- xperts needed.
- rade-off between number and quali y of nnotations
- ubjectivity and disagreement betwe n xperts.
- here is no "**ground truth**".



7

If we're trying to create something like a self-driving car, we can use something like this: captchas. If you've been on the internet in the past decade, you've probably encountered one of those. They don't just check if you're a robot or a human: the input you provide is used to train and refine deep learning models. By crowdsourcing the annotation of the data, it's easier to get larger datasets.
But in digital pathology, you can't do that. If you ask thousands of random internet users to select all the mitotic cells in an image, you will not get very useful annotations. In digital pathology, the annotations are complex, and it requires trained experts. There will always be a trade-off between the number of annotations you can get, and the quality of these annotations. There is also a certain amount of subjectivity and disagreement between experts, as the criteria used by pathologists can be very difficult to evaluate.

Impact of real-world annotations on the training and evaluation of deep learning algorithms in digital pathology

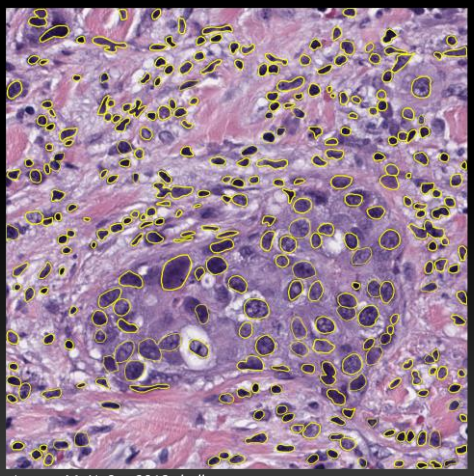


Image: MoNuSeg 2018 challenge

Study of **imperfect annotations**, their effect on **deep learning** algorithms, and the **learning strategies** we can use to counteract those effects.

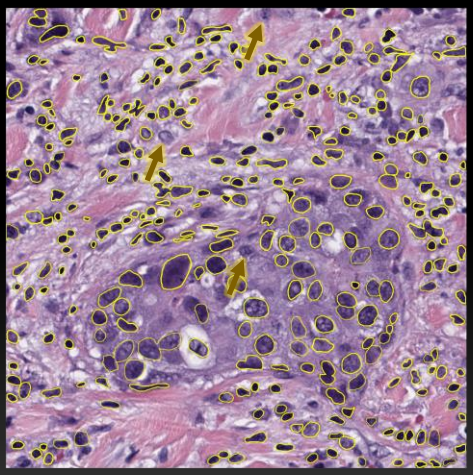
Analysis of **evaluation processes**, and how imperfect annotations and inter-expert disagreement can be better taken into account.

Review of **digital pathology competitions**, how they (don't) take these problems into account, and how they can be improved.

8

So that brings us back to the title of the thesis: what is the impact of the reality of the annotations we can get in digital pathology on the training of deep learning algorithms, and also on their evaluation.
There are three big topics that I want to cover in the rest of this presentation, and for each I will show some examples of our experiments, and explain the conclusions that we were able to reach.
The first is the study of imperfect annotations and their effect on the learning process. The second is the analysis of evaluation processes and metrics, and how imperfect annotations and inter-expert disagreement can be taken into account better. Finally, I will talk about our review of digital pathology competitions, how they take, or don’t take, these questions into account, and how they can be improved in the future.

Imperfect annotations Evaluation of algorithms Competitions



Incomplete annotations (missing annotations)

Imprecise annotations (uncertainty on the boundaries)

Incorrect annotations (uncertainty on the labels)

Our **contributions:**

- Evaluate the **effects** of imperfect annotations on deep learning models.
- Find adapted **learning strategies** to counteract those effects.

9

Let's start with imperfect annotations. This is an image from the MoNuSeg 2018 nuclei segmentation competition, so the goal is to find the nuclei, which are contoured here in yellow. This image is part of the test set, meaning it's used to compare the results of competing algorithms to the expected output. In this image, however, there seem to be some objects which are *probably* nuclei as well, but are not annotated. This is a very common type of imperfection.

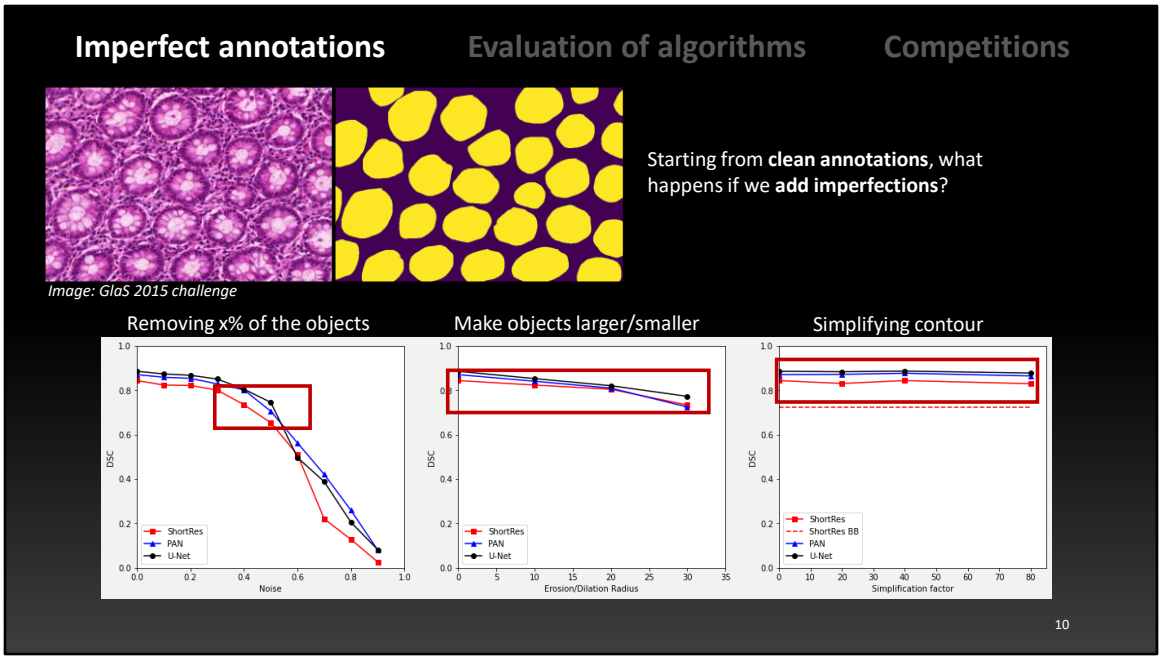
Digital pathology often deals with very large images, and we can have very small and numerous objects of interests. It's therefore very common to have **incomplete annotations**, where only some of the images, or some regions of the images, are annotated.

There are other types of imperfections as well.

The shape of the objects can be fairly complex. For segmentation problems, making pixel-precise annotations is extremely time-consuming. So it's common to have **imprecise annotations**, where there is an uncertainty on the exact boundaries.

Finally, when humans have to do a very repetitive task, like annotating thousands of nuclei in an image, there will always be some plain mistakes. **Incorrect annotations** means that there is an uncertainty on the labels, even in the annotated regions.

We studied the effect of imperfect annotations, as well as some learning strategies that can be used to counteract those effects.



We started with some very good, “clean” datasets, such as this one from a colorectal cancer gland segmentation challenge. The original annotations are as close to “pixel perfect” as we can get in a digital pathology dataset.
So what we did was to artificially add imperfections. We simulated different types. For missing annotations, we randomly removed annotated objects from the dataset. We randomly deformed the shape of the objects by making them smaller or larger. And we simplified the contours, simulating a quicker annotation method where, instead of precisely going around the contour, the expert would just have selected a few points to give us a polygonal approximation.
We trained three different deep learning models on corrupted datasets with different levels of corruption. Our results show that there is a certain natural robustness to some imperfections. In particular, the models are very robust to contour simplification, even when only four or five points remain in the contour. Removing too many objects, however, lead to a strong performance drop.

Imperfect annotations Evaluation of algorithms Competitions

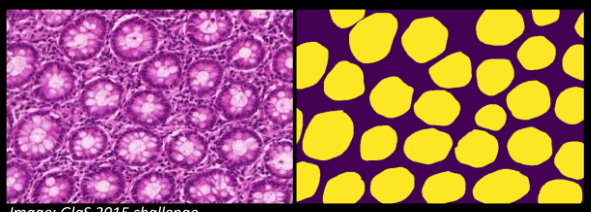
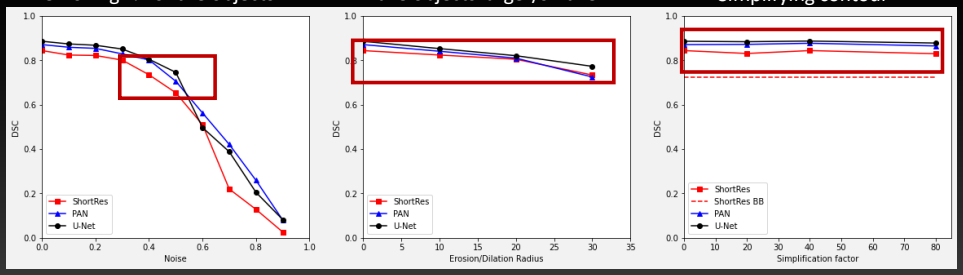


Image: GlaS 2015 challenge

Best strategy to annotate a digital pathology dataset?

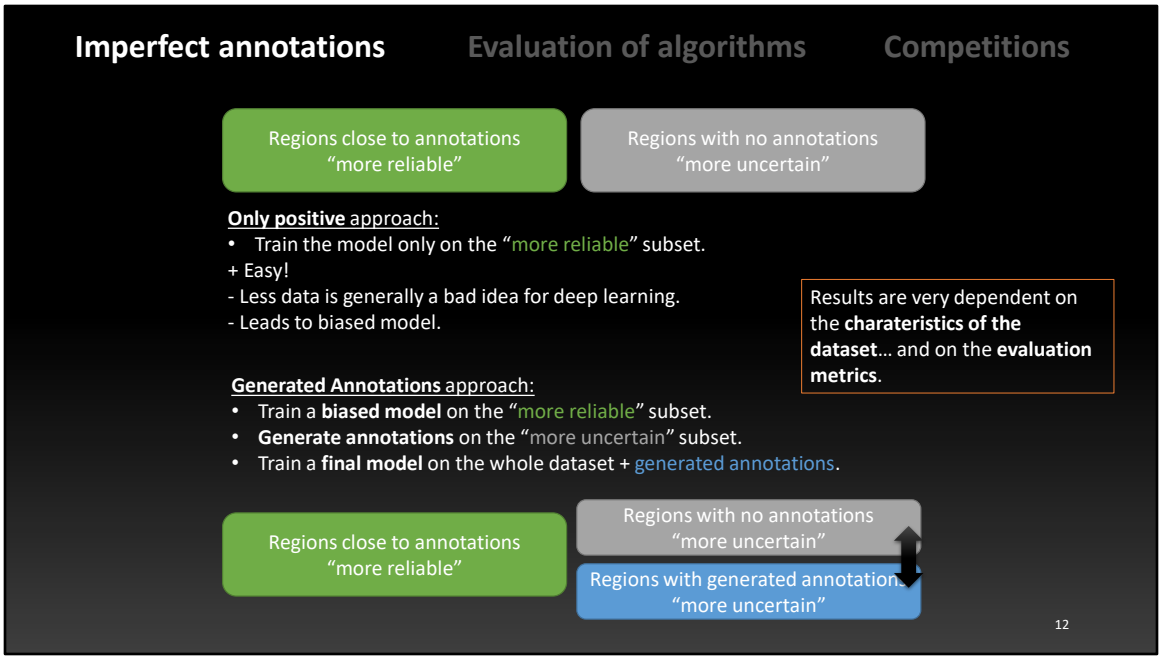
- Don't worry too much about the contours.
- More annotations (with correct labels) > precise annotations

Removing x% of the objects | Make objects larger/smaller | Simplifying contour



11

This indicates that, when annotating datasets for deep learning models, the best strategy for the expert pathologist could be to provide a rough set of simplified annotations, while making sure that few objects are missed.



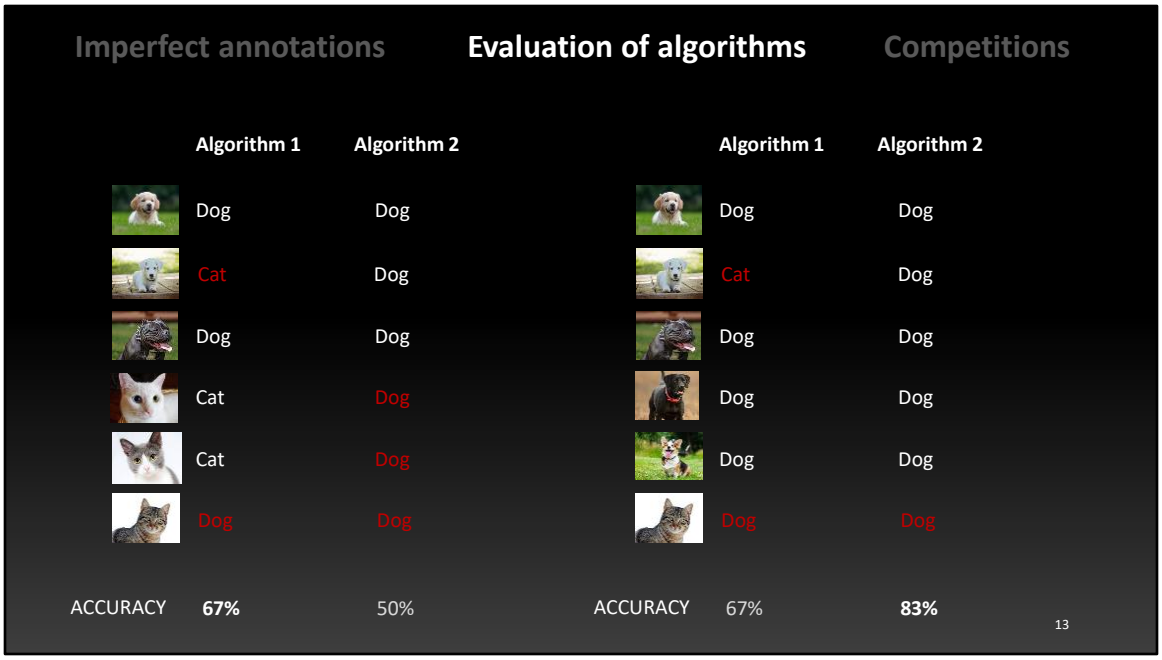
What kind of learning strategies can we use? Well, as we've found, the main problem we have is the missing annotations. So, in this type of datasets, we make the assumption that the regions where we have annotations are generally **more reliable** than the regions with empty annotations. In other words: it's more likely for an expert pathologist to miss an object than to randomly annotate something where there is nothing.

One easy thing that we could do is therefore to only use the "more reliable" part of the dataset, and to discard the rest. It's easy, but it also means that we don't use all of the data. And, as we've said before, deep learning algorithms need to be fed as much data as possible to work well. It also inevitably leads to a biased model, which will tend to predict objects everywhere.

The method that worked best according to our experiments tried to solve this issue and to use the whole data despite the uncertainty. The way we did that was to first train a biased model as with the "only positive" approach. Then, we use this model to generate annotations on the regions with no annotations. The final model is trained using all the data but, for the regions with no annotations, we randomly switch between using the original, empty annotations, and the generated ones. This allows us to use all the data, while recognizing the uncertainty of the annotations.

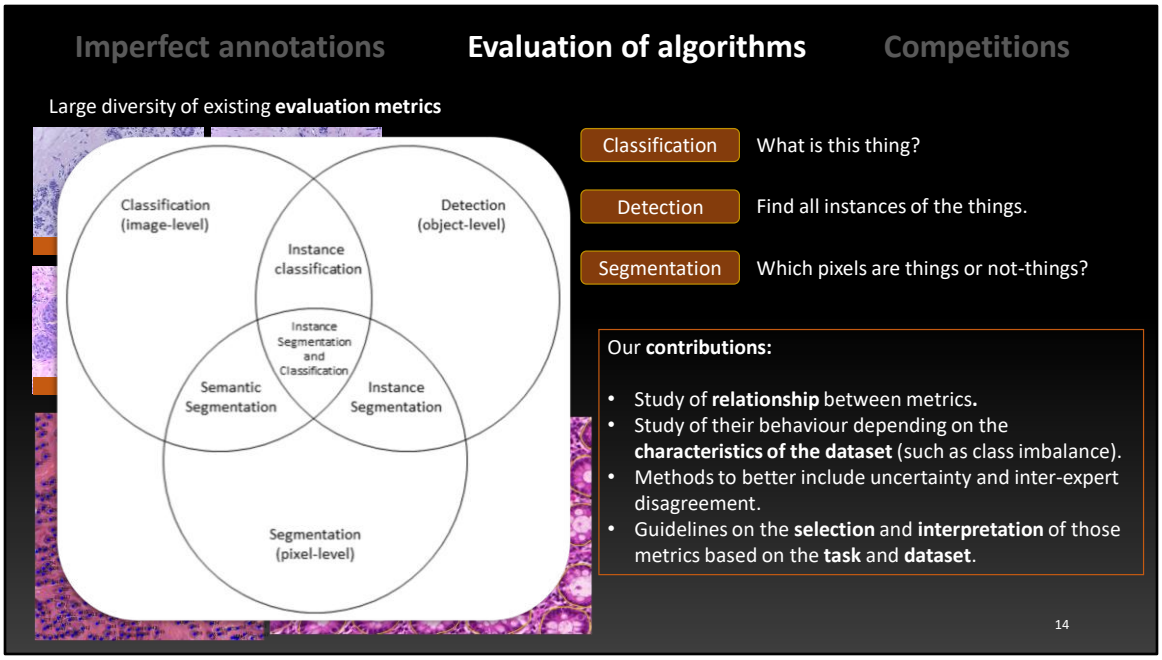
We also noted, however, that these results were highly dependent on the

characteristics of the dataset… and also on the metric we used to evaluate the results.

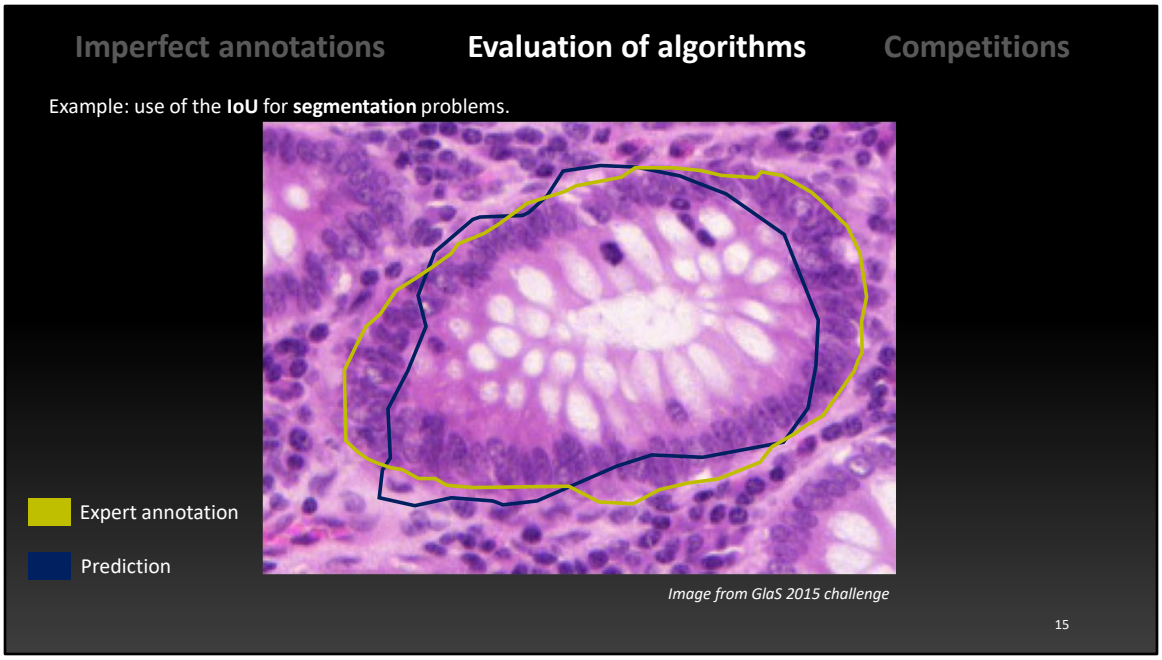


This leads us to our next topic: how to evaluate an algorithm?
It may seem like a fairly easy question. We see how many times they are right, how many times they are wrong, that's it. But I want to quickly show why that's not always the case.
Let's take a task that has nothing to do with digital pathology. Let's say that we are comparing two algorithms on their ability to classify images between cats and dogs. We test the first algorithm on our evaluation set, and find that it has an accuracy of two third, or around 67%. It's not a very good algorithm, but it seems to have learned something about the data. The second algorithm, meanwhile, is really bad. It always predict "dog". In our evaluation set, it leads to an accuracy of 50%. In this case, the algorithms are well ranked: the first algorithm, while not very good, is certainly more useful than the second.
But what if we change our evaluation dataset a little bit? There is, after all, nothing that guarantees that a dataset will have exactly as many examples of each class. In real-world scenarios, in fact, that's very unlikely. We compare our algorithms again, and we still have the same score for the first one. But for the second, now, we get an accuracy of 83%! If we use the accuracy as a metric on this test set, we have to conclude that algorithm 2 is better than algorithm 1, even though it's functionally useless.

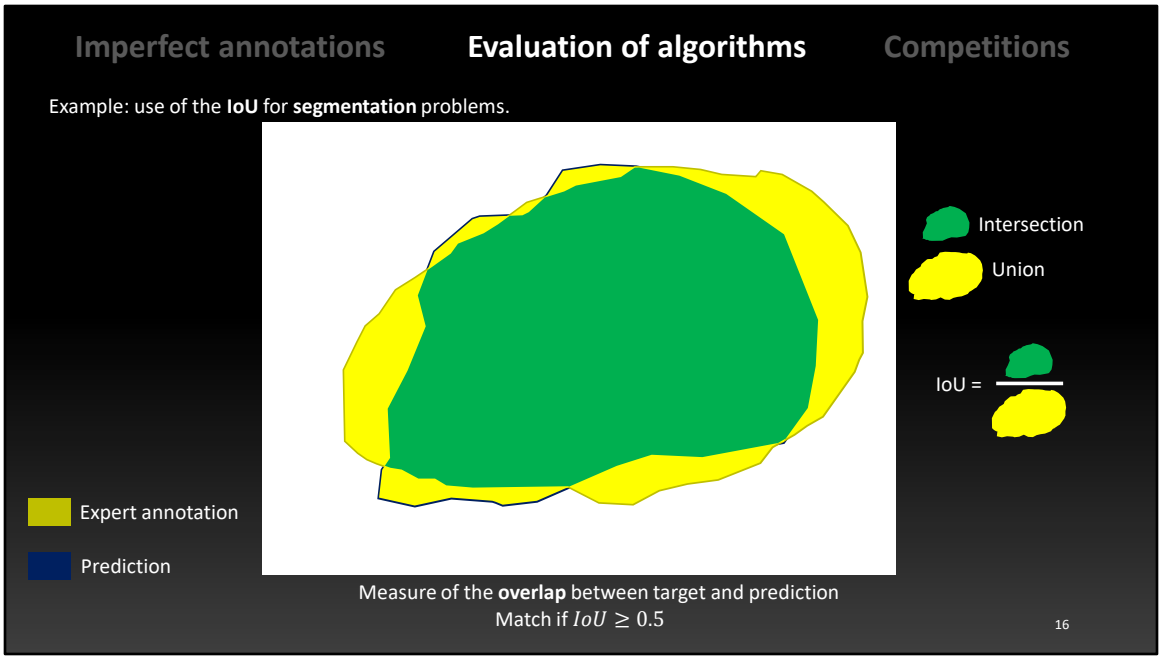
This kind of situations happen all the time in medical datasets, where, for instance, more invasive tumors tend to be less represented in the data than benign tumors. An algorithm that never recognizes an invasive tumor, however, would not be very useful, even though it may have a better accuracy than a model that doesn't always give such an optimistic prediction.



So what can we do? Well, fortunately, there are many evaluation metrics available to us. These metrics are generally related to the definition of the image analysis task.
In our work, we surveyed digital pathology competitions and publications, and we categorized the most common image analysis tasks using three different basic tasks: classification - “what is this thing?”, detection - “find all instances of things”, and segmentation - “find which pixels are things or not-things”.
In each of these basic tasks, there are many different commonly used metrics. More complex tasks can generally be defined as combinations of the three basic ones, and similarly more complex metrics tend to take the form of combinations of these basic metrics.
In the thesis, we studied the relationship between the different metrics: which ones tend to behave in the same way, and which ones can sometimes give very different results. We also studied their behavior depending on the characteristics of the data, such as the class imbalance that we showed in our cat and dog example, and how to better include the uncertainty and the differences of opinions between experts. From this, we tried to provide some guidelines on the selection and interpretation of those metrics.



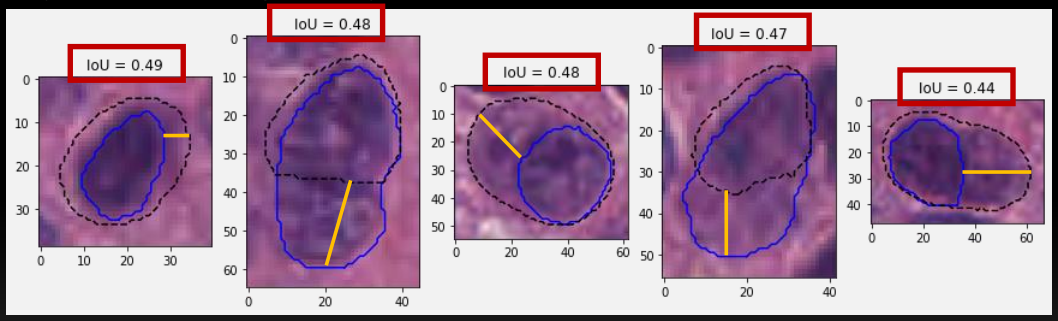
I want to show a few examples of the analyses that we made. The first example concerns the use of a segmentation metric called the “intersection over union”. This is one of the most commonly used metrics for this task, and it’s a relatively intuitive metric. Let’s say that we have an image, with an expert annotation so that we know which pixels belong to the object. Now we want to evaluate a prediction from an algorithm. How do we do that?



First, we take the intersection between the two objects, and compute the area.
Then, we take the union of the two objects, and also compute the area.
Finally, to get the “intersection over union”, we divide… the intersection by the union.
Not very suprising. This gives us a fairly straightforward measure of the overlap between the two objects. The IoU will be equal to 1 if the objects are perfectly overlapping, and equal to 0 if they are completely separate.
So what can go wrong?

Imperfect annotations **Evaluation of algorithms** Competitions

Example: use of the **IoU** for **segmentation** problems.



Blue = expert annotation, Black = prediction. Based on results from MoNuSAC 2020 challenge

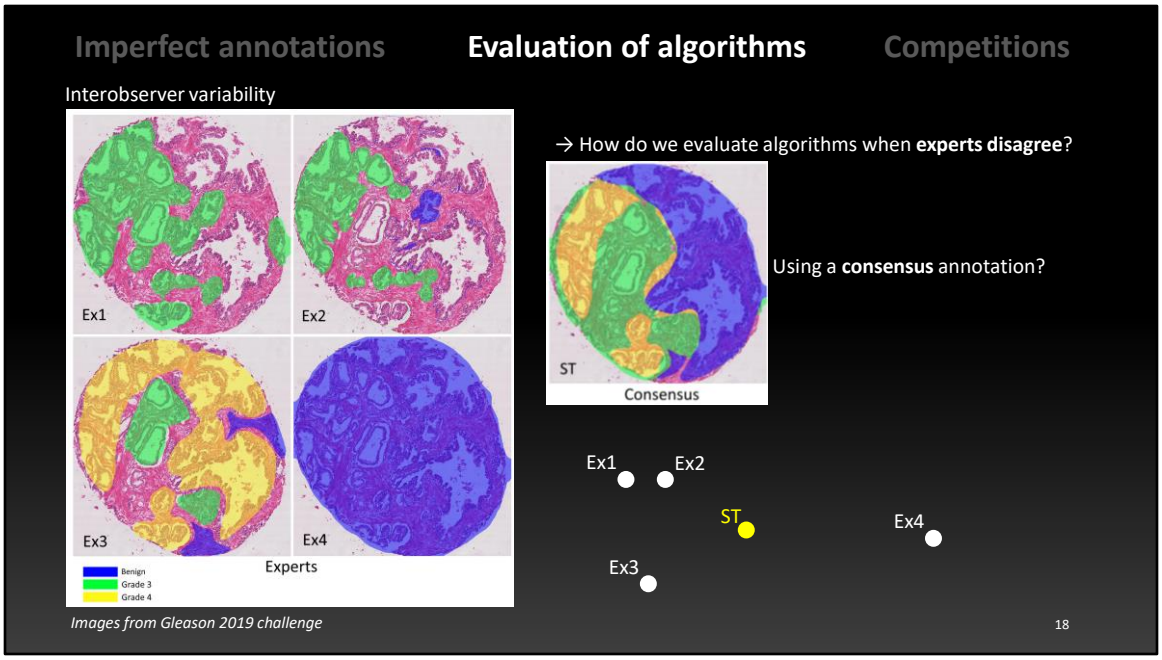
→ IoU alone is a bad metric for **small objects,** especially with **uncertain boundaries**…
yet is widely used for evaluation of nuclei segmentation!

→ Alternative(s) : uncertainty-aware IoU, metrics based on contour distance (e.g. Hausdorff's Distance)…
compute multiple metrics for richer insights.

17

Look at these five examples from the MoNuSAC 2020 challenge, with the expert annotations in solid blue and predictions from the algorithm of one of the teams that participated in the challenge in dashed black.
According to the IoU, these five images all show very similar segmentation performances. In all five cases, this performance is also relatively bad: an IoU under 0.5 is often interpreted as meaning that the object wasn't actually detected by the algorithm. Visually, however, it's clear that it's not the case here. The first image in particular show a segmentation that's actually very good. The shape of the object is correct, and the predicted and annotated boundaries are both in the same fuzzy region that surrounds the nucleus. The other segmentations are clearly not as good, despite their similar IoU.
The problem here is that the IoU is bad for small objects with uncertain boundaries. The central region of the nuclei, where the annotation is more certain, has a very small area, which means that any error on the border region will have a much larger impact.
Some alternatives are to use a slightly modified versions of the metric, which doesn't penalize errors that are only a few pixels away from the annotated border. Another possibility is to use a metric such as Hausdorff's Distance, which measures the distance between the contours, and in this case gives us more information on

whether the correct shape was found.
In general, however, different metrics will have their pros and cons, and provide different information, so it's always advisable to compute several different metrics to get richer insights on the performances of the algorithms.



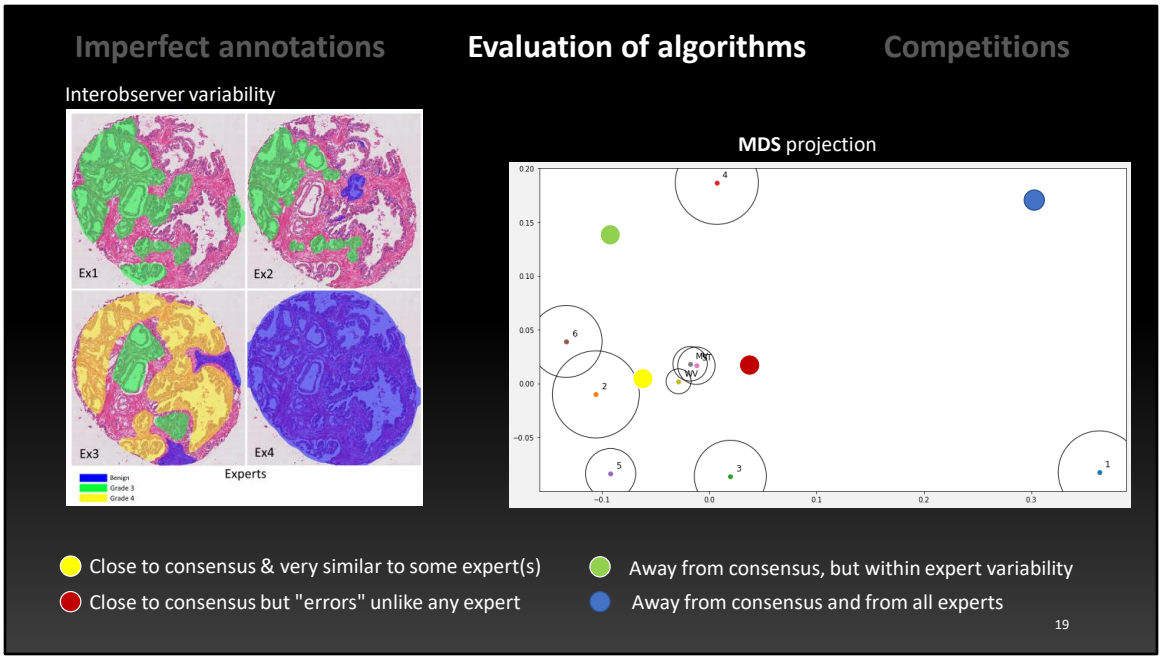
Let's now look at another example. What we have here is an image from the Gleason 2019 challenge, with the annotations made by four different expert pathologists. The task is to find Gleason patterns, which are indicative of how invasive the tumor is. The colors here show different grades. We can immediately see that, for this particular image, the opinions of the experts vary a lot, with one of them showing everything as benign - in blue - and others finding patterns of different grades. So how do we evaluate algorithms when experts disagree?

A common strategy is to use a **consensus annotation**. This can either be done by putting all the experts in the same room and ask them to find an agreement, or, as was done in this particular challenge, by using an automated method to merge the different annotations together. Then, the results of different algorithms can be compared to the consensus.

The main problem with this approach is that the consensus no longer contains the information about the uncertainty of the annotations, so there is no difference between images where all experts were in agreement and images like this one where they disagreed a lot.

What we proposed in our work is to keep the individual expert annotations, and to compare them with each other, and with the consensus. We can look at the similarity between experts as a form of distance. So for instance here, expert 1 and 2 share

many similarities in their annotations, so they should be relatively close together. Expert 3 has some parts in common, but is clearly further away. Expert 4 is even further away, with their “all benign” annotation. The consensus would fall somewhere in the middle.



We applied this idea to the entire Gleason 2019 dataset, to create a visualization of the expert disagreement, using a method called “Multi-Dimensional Scaling”, or MDS. There are six experts in total, shown here with the labels 1 to 6, and we computed three different consensus methods, which appear here in the middle. The first interesting thing in this approach is that we can see how the experts relate to each other, and identify those, like expert 1 here, that are often in disagreement with the others.

But this can also provide some interesting insight on the performances of algorithms. If, instead of computing a metric on the consensus, we compute it on all experts and all consensus methods, we can then see where the algorithm would appear on the visualization. This can allow us, for instance, to differentiate between an algorithm that’s close to the consensus and very similar to some of the experts, from an algorithm that may be just as close to the consensus but with disagreements that are very unlike any of the experts. Similarly, we could differentiate between algorithms that are further away from the consensus, but still within the range of interexpert variability, from algorithms that are away from the consensus *and* from all experts.

Imperfect annotations | **Evaluation of algorithms** | Competitions

Metrics for complex tasks

Nuclei instance segmentation and classification

Detect nuclei
Objects

Segment nuclei
Pixels

Classify nuclei
Epithelial
Lymphocyte
Neutrophil
Macrophage

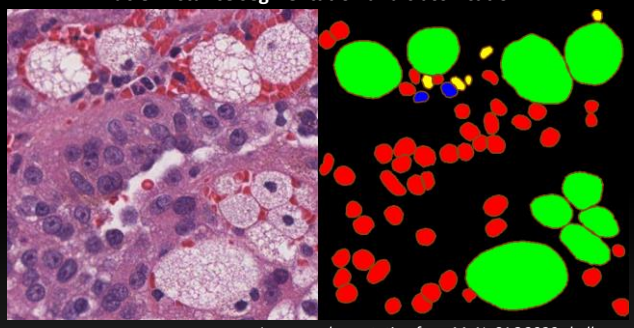


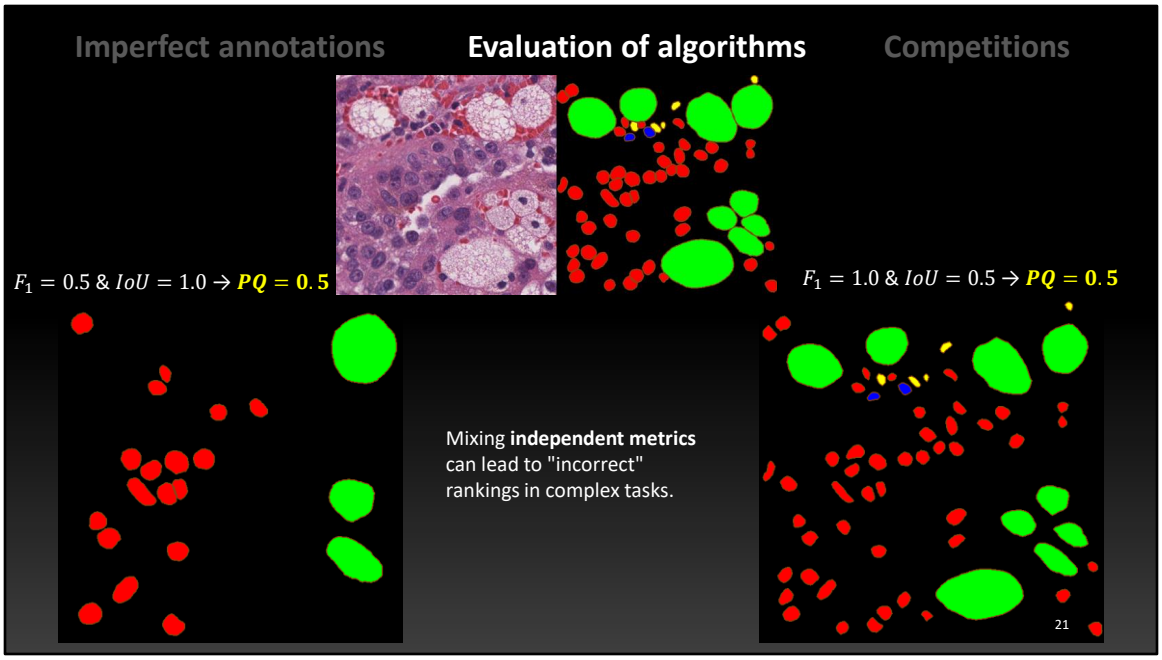
Image and annotation from MoNuSAC 2020 challenge

$$PQ = DetectionQuality \times SegmentationQuality$$
$$PQ = F_1 \times IoU$$

(computed per-class)

20

This idea of maximizing the insights we can get on the algorithms is something that we came back to a lot in our work. It also applies to the evaluation of complex tasks, such as this one from the MoNuSAC challenge. Here, we need at the same time to detect the individual nuclei objects, to segment them at the pixel level, and to classify these nuclei in four different categories.
The metric used in the challenge was the Panoptic Quality. The Panoptic Quality combines two of the basic metrics, the detection F1 score and the average segmentation IoU of the detected nuclei, by multiplying them together. One of the problems here is that the interpretation of range of values for those two metrics is very different.

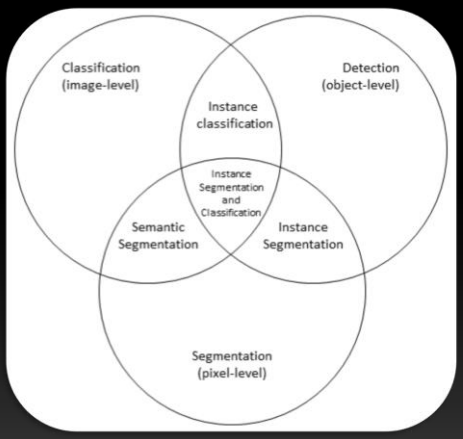


We can use two fabricated example predictions to illustrate what can go wrong.
As we see on the left, a PQ of 0.5 can be obtained with an F1-score of 0.5 and an IoU of 1, which corresponds to an algorithm that misses a very large portion of the nuclei, making it completely useless in practice.
On the right, we also have a PQ of 0.5, but this time with an IoU of 0.5 and an F1-score of 1. An IoU of 0.5, however, can correspond to a segmentation that's relatively good for small nuclei. This prediction is therefore actually very good: it correctly detects all the objects, with a decent segmentation.
It is therefore very weird to say that an algorithm with an F1 score of 0.5 and an average IoU of 1 has the same performance as an algorithm with a F1 score of 1 and an average IoU of 0.5.
Mixing independent metrics, in this case, can clearly lead to "incorrect" rankings in complex tasks.

Imperfect annotations **Evaluation of algorithms** Competitions

What we **recommend:**

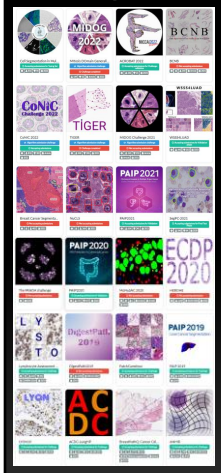
- **Simple metrics**, ranked separately, to **compare and contrasts** methods on all their strengths and weaknesses.
- Compare to **all available experts** rather than on a single "ground truth".
- **Embrace the uncertainty** if possible. Having ex aequos is not a problem.



22

So this is what we recommend based on our analyses of evaluation metrics:
Use simple independent metrics, don't try to combine them into something that becomes impossible to interpret.
If multiple experts are available, and ideally they should be, compare algorithms to all of them rather than trying to find a single "ground truth", which probably doesn't exist anyway.
Finally, there will be uncertainties on the value of the metric, so we shouldn't pretend that they are absolutely right. If two algorithms are within the uncertainty in the annotations, then we cannot say that one is better than the other. Having ex aequos may not be the most satisfying outcome, but it's better than discarding a method that is as promising as a superficial winner.

Imperfect annotations



Evaluation of algorithms

Challenge	Target	Metric(s)
PR in HIMA 2010	Centroblasts in follicular lymphoma.	REC, SPE[30]
MITOS 2012	Mitosis in breast cancer.	**F1**, PRE, REC
AMIDA 2013	Mitosis in breast cancer.	**F1**, PRE, REC
MITOS-ATYPIA 2014	Mitosis in breast cancer.	**F1**, PRE, REC
GlaS 2015	Prostate glands	**F1**
TUPAC 2016	Mitosis in breast cancer.	**F1**
LYON 2019	Lymphocytes in breast, colon and prostate.	**F1**
DigestPath 2019	Signet ring cell carcinoma.	PRE, **REC**, **FPs per normal region**, **FROC**[31].
MoNuSAC 2020	Nuclei	**F1** (as part of the PQ)
PAIP 2021	Perineural invasion in multiple organs.	**F1**
NuCLS 2021	Nuclei in different organs.	AP@.5[32], mAP@.5:.95
MIDOG 2021	Mitosis in breast cancer.	**F1**
Conic 2022	Nuclei	**F1** (as part of the PQ)

Challenge	Classes	Metric(s)
Brain Tumour DP 2014	Low Grade Glioma / Glioblastoma	**ACC**
MITOS-ATYPIA-14	Nuclear atypia score (1-3)	**ACC** with penalty[33]
BIOIMAGING15	Normal, benign, *in situ*, invasive	**ACC**[34]
TUPAC16	Proliferation score (1-3)	κ_q
CAMELYON16	Metastasis / No metastasis	$AUROC_{metastasis}$
BACH18	Normal, benign, *in situ*, invasive	**ACC**
C-NMC19	Normal, malignant	**Weighted sF1**[35]
Gleason 2019	Gleason grades	κ (included in a custom score)
PatchCamelyon19	Metastasis / No metastasis	$AUROC_{metastasis}$
DigestPath19	Benign / Malignant	$AUROC_{malignant}$
HeroHE20	HER2 positive / negative	$F1_+, AUROC_+, SEN_+, PRE_+$
PANDA20	Gleason group (1-5)	κ_q
PAIP20	MSI-High / MSI-Low	$F1_{High}$
NuCLS21	Different types of nuclei	$ACC, MCC, \mu AUROC, MAUROC$

Competitions

Challenge	Target	Metric(s)
PR in HIMA 10	Lymphocytes	DSC, IoU, HD, MAD
Brain Tumour DP 14	Necrosis region	**DSC**
GlaS 2015	Prostate glands	**DSC, HD**
SNI 15-18	Nuclei	**DSC**[36]
MoNuSeg 18	Nuclei	**IoU**[37]
BACH 18	Benign / in situ / invasive cancer regions	**Custom score**
Gleason 19	Gleason patterns	**DSC**[38] (included in a custom score)
ACDC@LungHP 19	Lung carcinoma	**DSC**
PAIP 19	Tumour region	**IoU**
DigestPath 19	Malignant glands	**DSC**
BCSS 19	Tumour / stroma / inflammatory / necrosis / other tissue segmentation.	DSC
MoNuSAC 20	Nuclei (epithelial / lymphocyte / neutrophil / macrophage)	**IoU** (as part of the PQ)
PAIP 20	Tumour region	**IoU**
SegPC 21	Multiple myeloma plasma cells	**IoU**
PAIP 21	Perineural invasion	HD
NuCLS 21	Nuclei (many classes)	IoU, DSC
WSSS4LUAD 21	Tumour / stroma / normal tissue	**IoU**
CoNIC 22	Nuclei (epithelial / lymphocyte / plasma / eosinophil / neutrophil / connective tissue)	**IoU** (as part of the PQ)

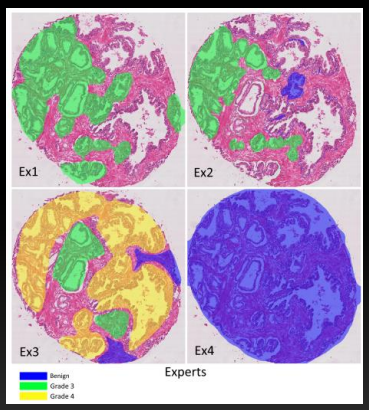
Our **contributions:**

- Review of digital pathology challenges.
- Which **metrics** are used? How do they determine the "**ground truth**" (and handled expert disagreement)?
- What are the trends in the **top methods**?

23

The last topic that I want to briefly talk about is that of digital pathology competitions. I have used many examples from different competitions through this presentation, and through the thesis. Competitions have had a very large impact on digital pathology research in the past decade, and are very much linked to the shift towards deep learning in image analysis.
So we wanted to have a closer look at how they were organized, and how they dealt with real-world annotations, and evaluation processes.
We reviewed digital pathology challenges organized between 2010 and 2022. Based on the publicly available information on those challenges, we focused on which evaluation metrics they used for which task, how they determined what the “ground truth” was for their evaluation, and how they handled inter-expert disagreement. We also looked at the trends in the top ranked methods.

Imperfect annotations | Evaluation of algorithms | **Competitions**



Many **challenges** use a **single expert** for their annotations.

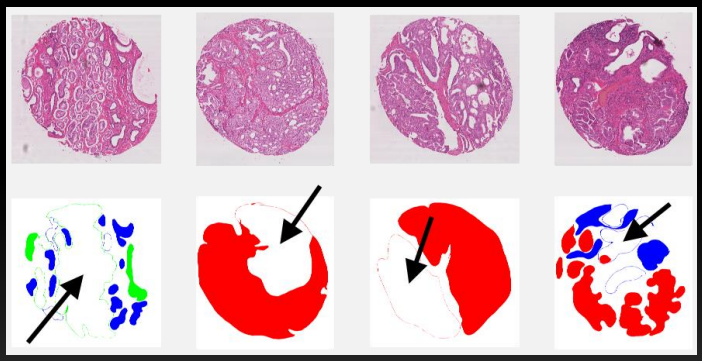
Several use **multiple experts** in an **informal consensus** (only the consensus is available, often few information on the process).

Only **1/21** reviewed segmentation challenge provided **individual annotations** from **multiple experts**.

24

One of the things we noted was that many challenges use a **single expert** for their annotations, and completely ignore the potential disagreements. Those that use multiple experts typically use an informal consensus method, with few explanations on how exactly they came to an agreement, and no annotation released beside the agreement.
Of the 21 segmentation challenges which we reviewed for this particular aspect, only one - the Gleason 2019 challenge - provided individual annotations from all experts. This is obviously a bit worrying.

Imperfect annotations Evaluation of algorithms **Competitions**



Mistakes in annotation maps (Gleason 2019)

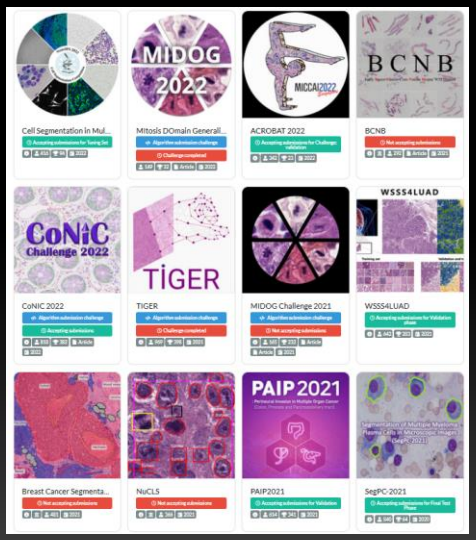
Challenges are **complex** to organize.
Problems found in…
… the constitution of the dataset
… the annotations provided
… the evaluation code

These mistakes were found in challenges **with higher than usual transparency**.
Finding mistakes is part of the scientific process!

25

More worrying are the different mistakes that we found while reviewing the challenges. Now I want to be clear here that my point is not to say that the challenges where we found those mistakes are bad. The point is that challenges are extremely complicated to organize, and that mistakes can easily appear through the whole process.
We found problems starting from the constitution of the dataset itself. We also found problems in annotations, like here with the Gleason 2019 dataset, where in many annotations the contours were not properly closed, leading to large regions of the image where the expert clearly intended to annotate a pattern, but the annotation doesn't contain any.
We also found errors in the evaluation code of another challenge, which had led to incorrect results being published.
The main thing that I want to emphasize here is that the reason we found those mistakes is not necessarily that those challenges were particularly badly organized, but rather that these are challenges that were particularly transparent in some aspects of their process, allowing us to better analyze their results. Finding mistakes is part of the scientific process. Making the mistakes hard to find is the problem.

Imperfect annotations Evaluation of algorithms **Competitions**



Our main **recommendations:**

- **Increased transparency** (release all elements needed for replicability)
- Recognize that **competitive** goal (finding "a winner") may clash with **scientific** goal (learning about the algorithms).
- **Keep pathologists in the loop**: in the **task definition**, in the choice of **evaluation metric(s)**, in the **interpretation of the results**.

26

This is why our first recommendation from our review of digital pathology challenges is that we need more transparency. Most challenges simply don't publicly release all the information needed to replicate their results, and to find potential errors.
The second thing we note is that there is often a tension in the evaluation methods between the competitive goal of the challenge, which is to find a winner, and the scientific goal, which is to learn more about the algorithm, and about the object of the task. The competitive goal may be necessary as an incentive, but it's important to at the very least not limit the evaluation to a metric designed for easy ranking.
Finally, we remarked on the importance of keeping pathologists in the loop through the process. From the task definition to the interpretation of the result, the pathologists can make sure that all the work done developing and training these huge models stay focused on things that actually help the pathologists and, at some point, the patients.

Conclusions

- **Imperfect annotations** impact the training of deep learning algorithms… but also their **evaluation**. Imperfections are not limited to the **training set**.
- The **choice of evaluation metric(s)** is not trivial. Analysing the **behaviour of the metrics** for a given dataset is necessary to correctly interpret the results.
- **Experts disagree**. We cannot just assume that "the truth is somewhere in the middle".
- **Digital pathology challenges** are very useful, drive our understanding of the state-of-the-art, and require a large amount of energy to organize… so we should make sure that their results are **reliable**, **replicable** and **reusable**!

Place of **AI image analysis**
in the **digital pathology workflow** ?

~~**Fully automated** diagnosis~~

Quantification, pointing to **regions/objects** of interest

Research and discovery of new biomarkers

…as a tool for pathologists, not instead of pathologists

27

After this little tour of some of the work done during this thesis, let's take a step back and look at what we learned, and where that leads us for the future of AI in digital pathology.
First, we have shown the impact of imperfect annotations on the training process of deep learning algorithm. We've also shown that this impact is also present in their evaluation, as imperfections are also part of the test sets.
Choosing the right metric, or metrics, is not easy. The characteristics of each dataset will have an impact on the behavior of the metrics, and it's always necessary to take the time to properly analyze that behavior so that the metric is correctly interpreted.
Third: experts disagree. That's fine: that's the reality of the medical world. But that means we cannot just assume that there is a "truth", and that it's somewhere in the middle. The diversity in expert opinions has to be a part of how we interpret the results of the algorithms.
We have studied many digital pathology challenges, and it is clear that they are an extremely useful resource in the domain. A large part of our current understanding of the state-of-the-art in digital pathology image analysis comes from these challenges. So it's really important to keep working on improving them, and making them more reliable, replicable and reusable.
One key question for the future is: what place can AI, at this point, take in the

pathology workflow.
It's clear that we are not ready yet for a fully automated diagnosis from digital pathology images. It's not even certain that it would really be something that we want in the first place. There is, however, a lot of potential for it to serve for more targeted tasks, like counting objects or measuring areas, or pointing to potential regions or objects of interest, always with an expert reviewing the results.
Another thing that we can do is use AI in places that are less sensitive, because we are not directly working on current patients. For instance: working on retrospective data, to try to find new relationships between the images and known patient outcome. These relationships can then be further explored by pathologists, to see if there may be some underlying biological explanations, and if that can help us better understand the disease.
And it's important to keep in mind that, despite all the hype that may exist around AI today, deep learning models are not intelligent entities. They are tools, and for the foreseeable future they will be tools that are best used by pathologists, to make them more efficient, better informed, and to help them take the best decisions with the patients. But modern deep learning methods are certainly no replacement for pathologists…

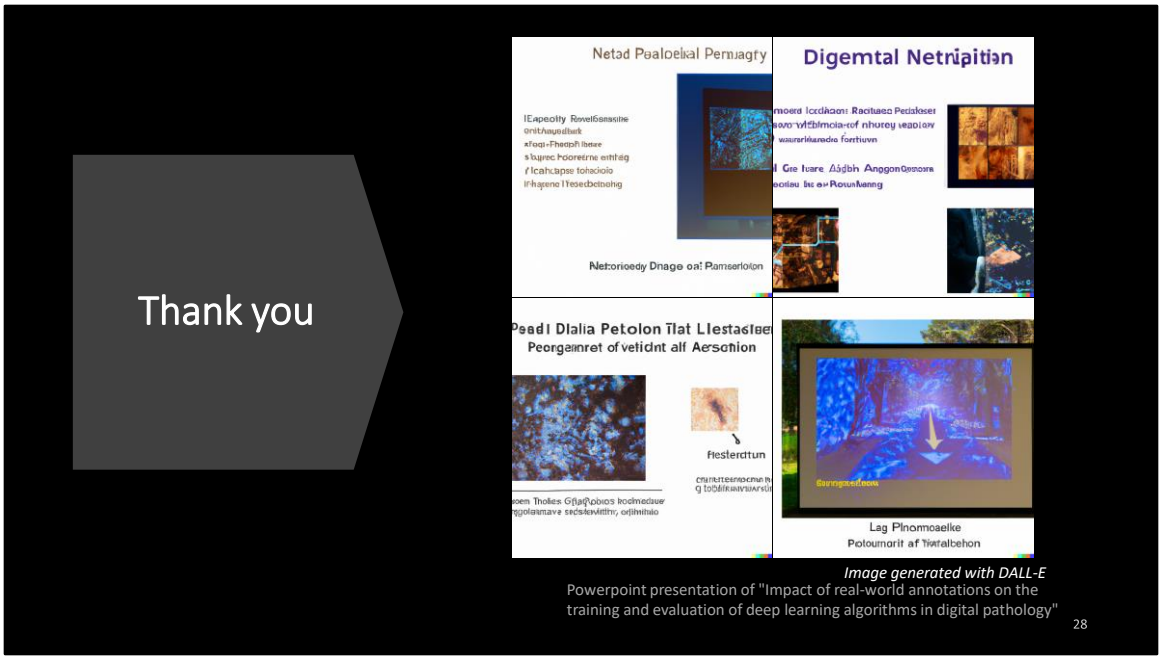


Image generated with DALL-E

Powerpoint presentation of "Impact of real-world annotations on the training and evaluation of deep learning algorithms in digital pathology"

28

…or for PhD students. I've tried to ask the DALL-E image generator to create the slides for this presentation, and I think my version is still a little bit better.
Thank you.