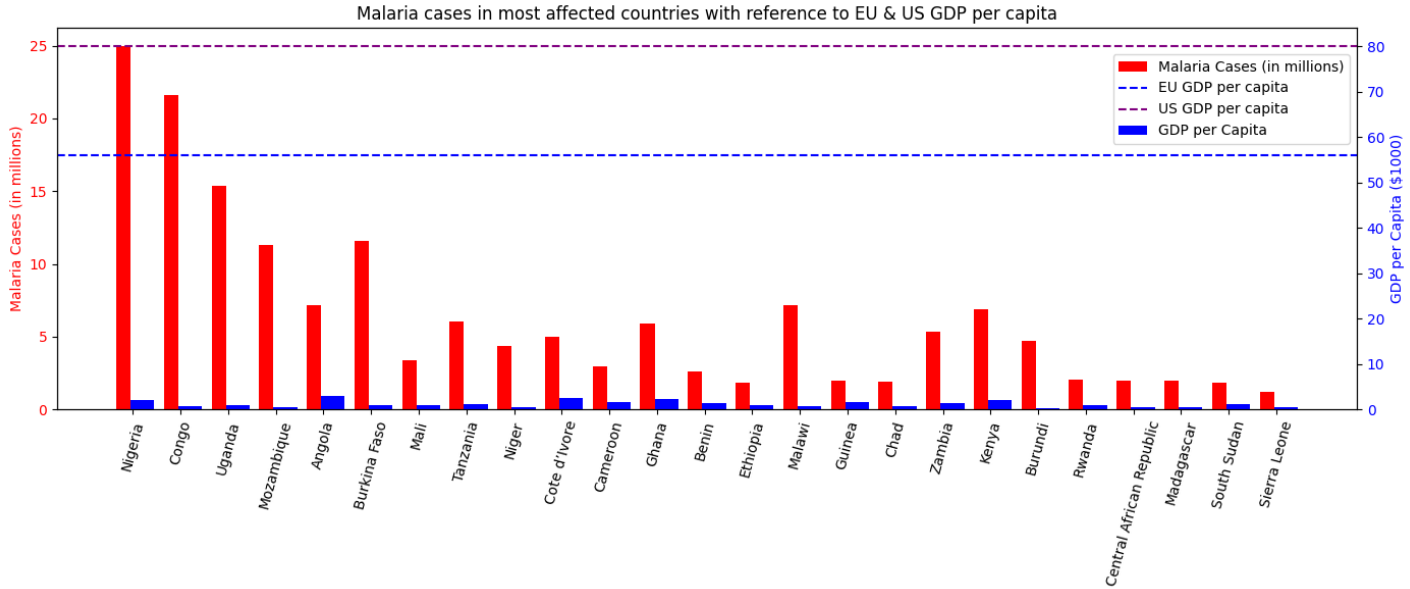
Malaria detection using computer vision

Capstone project for
Applied Data Science Program

Adam Leśniak

General problem statement

- There is a very serious problem of Malaria, there are estimates that **half a million people die yearly because of it.**



- Malaria is a problem of poor healthcare, therefore each real solution must be **extremely cost effective**
- It's not only 3rd world problem. There are 400000 emigrants that run away (affects rich EU) to escape misery in their own land

Word of caution

Lack of access to domain specialist means that everything here has to be critically reviewed and next iteration of this work will be required that includes better understanding of problem space.

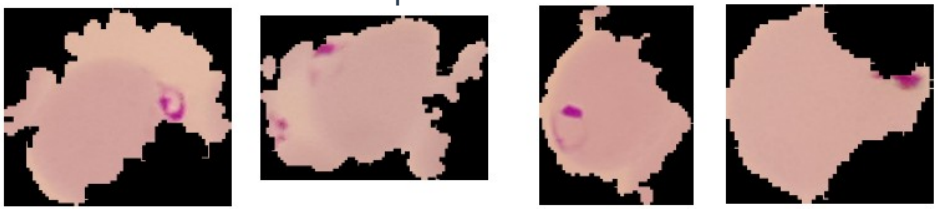
A little chaos was introduced by two conflicting goals. On one hand I should do the project and follow the logic that comes out of explorations, but on the other hand I was forced to do irrelevant stuff required by instructions. Next iteration of this work should focus on problem only.

Last, but not least I had to make presentation for unknown type of audience. Business, technical versatile managers? Each audience requires different approach... and me making presentation?!

Data

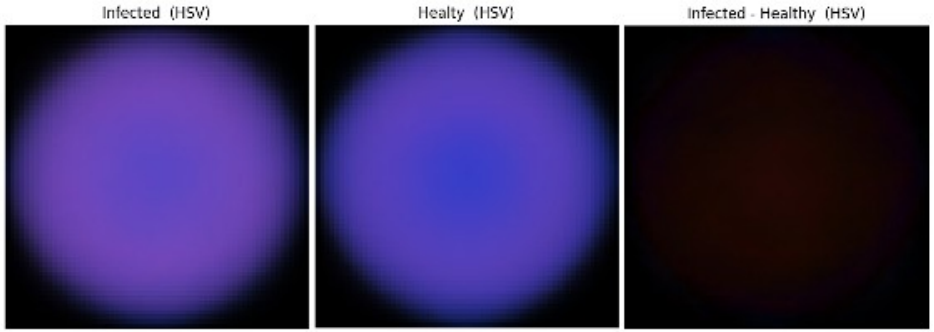
- Data clearly comes from thin blood smears, those are well separated images, separated according to their class: healthy and infected by malaria parasites
- Data is pretty good in terms of quantity (>27,5k images) and various – all kinds of angles, rotations, different lighting conditions, different shapes, different sizes, etc. equal amount of both categories – it's diverse and very easy to work with
- Vast majority of images is above 64x64 pixels
- Data is extremely precise – that may not translate well to real world taking into account price sensitivity

Few examples of infected cells



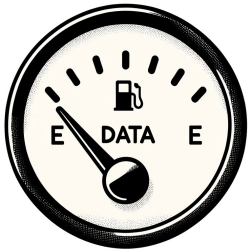
- It's clearly seen that it's not about shapes, so no stretching, reflections, rotations make sense, but it's about colors, so playing with them may make sense
- Colors should be easy to distinguish so we shouldn't need very complex model

Images means and difference between means



Requirements – sample size

- The thin blood smear area is assumed to be 10x20 mm.
- White blood cells are approximately 1% of blood volume, so we are working with data equivalent to 100x200 micrometers.
- One cell is about 15x15 micrometers, so that gives us about 80 cells in a sample.
- Malaria may infect less than 1% of cells, so such a number is not enough, now counting for optimizing for no false positives. Not even counting for probability that in normal distribution, even if we had 100 cells it's quite probable that by just pure chance there wouldn't be infected cell in sample of 100 cells, and we have 80!



Math doesn't work. It's impossible to do any real world application using this data.

We need to work with thick blood samples, as there is statistically to little data in thin blood sample to achieve meaningful accuracy on sample level.

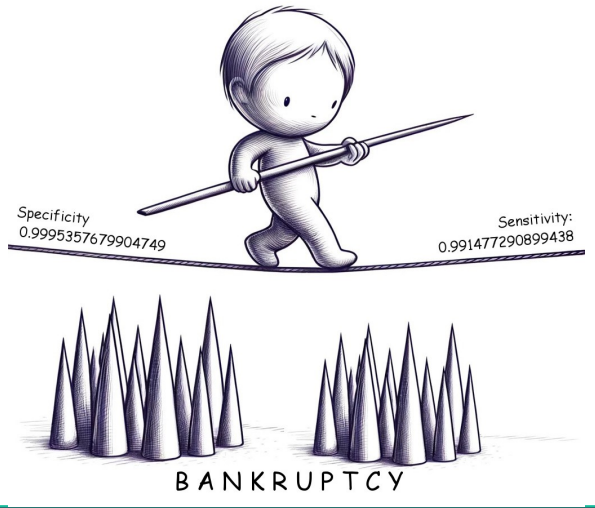
Requirements - accuracy

- Let's set the goal to achieve 95% sensitivity and 85% specificity – that's what's already achieved using different methods.
- That means in sample of 350 cells (usually for malaria detection samples of 200-500 cells are observed) **we have to have accuracy of 99.55% on single cell level**.
- That is hardly achievable, and even if we could achieve such results, that would be as good as already established methods, so who would go for unproven method that is no better than existing solutions?

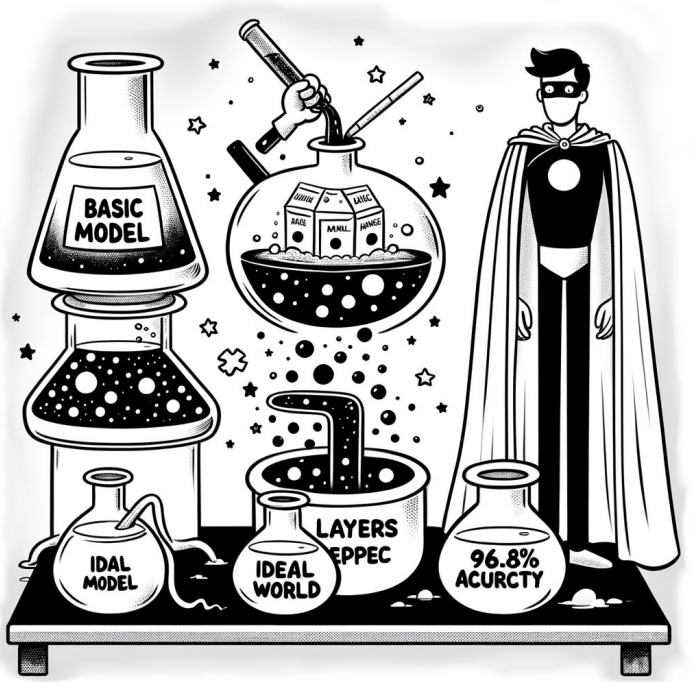
$$\text{Cell Specificity} = \text{sample specificity}^{\frac{1}{\text{cells in sample}}}$$

$$\text{Cell Sensitivity} = \text{sample sensitivity}^{\frac{1}{\text{cells in sample}}}$$

$$\textit{Sample Accuracy} = \frac{\textit{sample sensitivity} + \textit{sample specificity}}{2}$$



Models



How I proceeded:

- Did some basic very simple model to feel how data works with neural networks – it did well
- Improved that model slightly adding few more layers and got 96.8 accuracy
- Wasted time following statements from notebook to pursue models instead of working on promising one (at least won't waste your time presenting them)
- Finally run test on smaller resolution data and checked few mismatched images

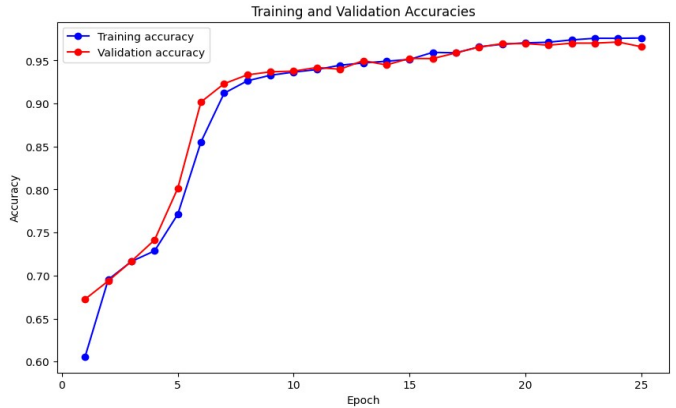
How I would proceed:

- Would tweak the model to get better accuracy (manipulate training data contrast, brightness and color shift to simulate different lighting conditions, play with hyper parameters, train form more epochs adding dropout layers if necessary)
- Would add data processioning to make model faster (remove value layer from HSV space, try to reduce number of kernels, try simpler activation functions in classification part)
- Would try working with little less accurate data – lower resolution to check if more affordable microscopes could do the job, as target is very price sensitive.

Best model and it’s potential

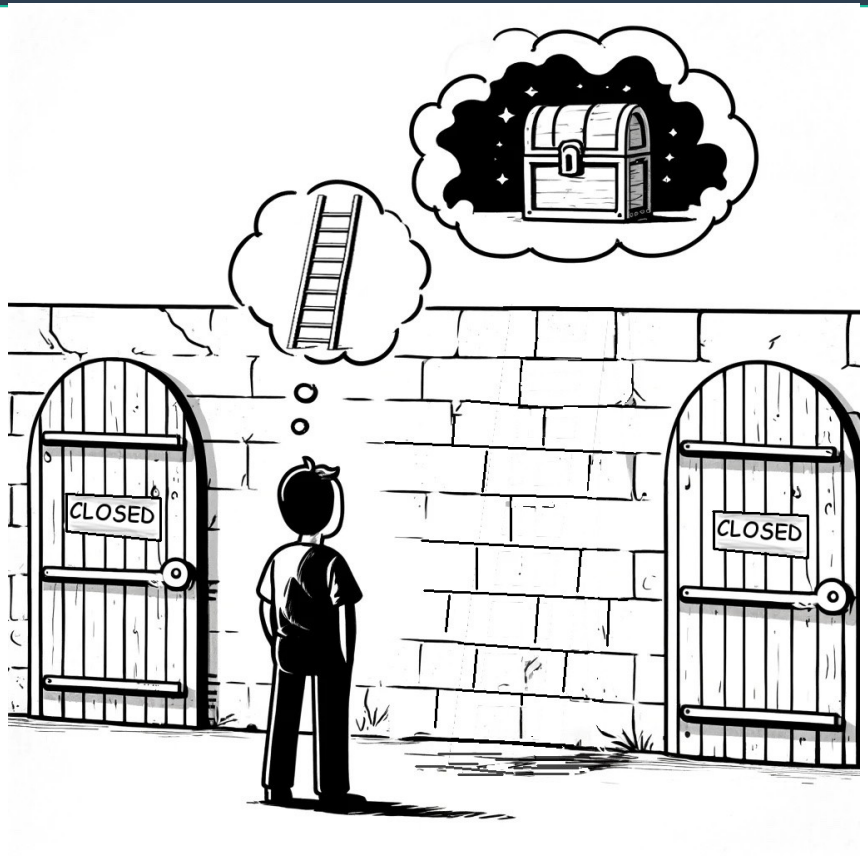
Second model was first real attempt and it was good fit with 96.8% accuracy
It’s relatively simple (**4 convolutional layers with relatively small kernel size**) what shows the data isn’t complex
After downsizing samples to **32x32 pixels** accuracy raised to 98.15%
After adjusting for mislabeled data, we can safely assume **accuracy over 99%**

Resizing(32, 32, 'bilinear'),
Conv2D(18, kernel=(3, 3), padding='same', input=(32, 32, 3)), LeakyReLU(0.25),
Conv2D(16, kernel=(3, 3), padding='same'), LeakyReLU(20),
Conv2D(12, kernel=(3, 3), padding='same'), LeakyReLU(0.15),
MaxPooling2D(pool_size=(2, 2)),
Conv2D(8, kernel=(3, 3), padding='same'), LeakyReLU(0.10),
Flatten(), Dense(32, activation='sigmoid'),
Dense(18, activation='sigmoid'),
Dense(1, activation='sigmoid')



- It has some room for improvement in computational requirements (reduce kernel size, add earlier pooling, padding to valid, remove value layer from input tensor...)
- It has quite a room for improvement in accuracy (add one more convolutional layer, maybe add one more Dense layer in classification layers, change predetermined LeakyReLU value to trainable PReLU, play with hyper parameters…)
- Fix labels in training and test data!
- **Don’t care about sensitivity vs specificity at research stage as they can be easily traded by adjusting threshold.**

Summary of experiment



- little math was enough that with this data it's no go for serious application
- we can provide solution that is almost as good as existing solutions, just untested
- why would anyone go for untested solution that is no better than existing ones in as sensitive area as human life?!
- playing just a little with deep learning shown great potential
- **we just need different data**

Solution – hybrid model idea

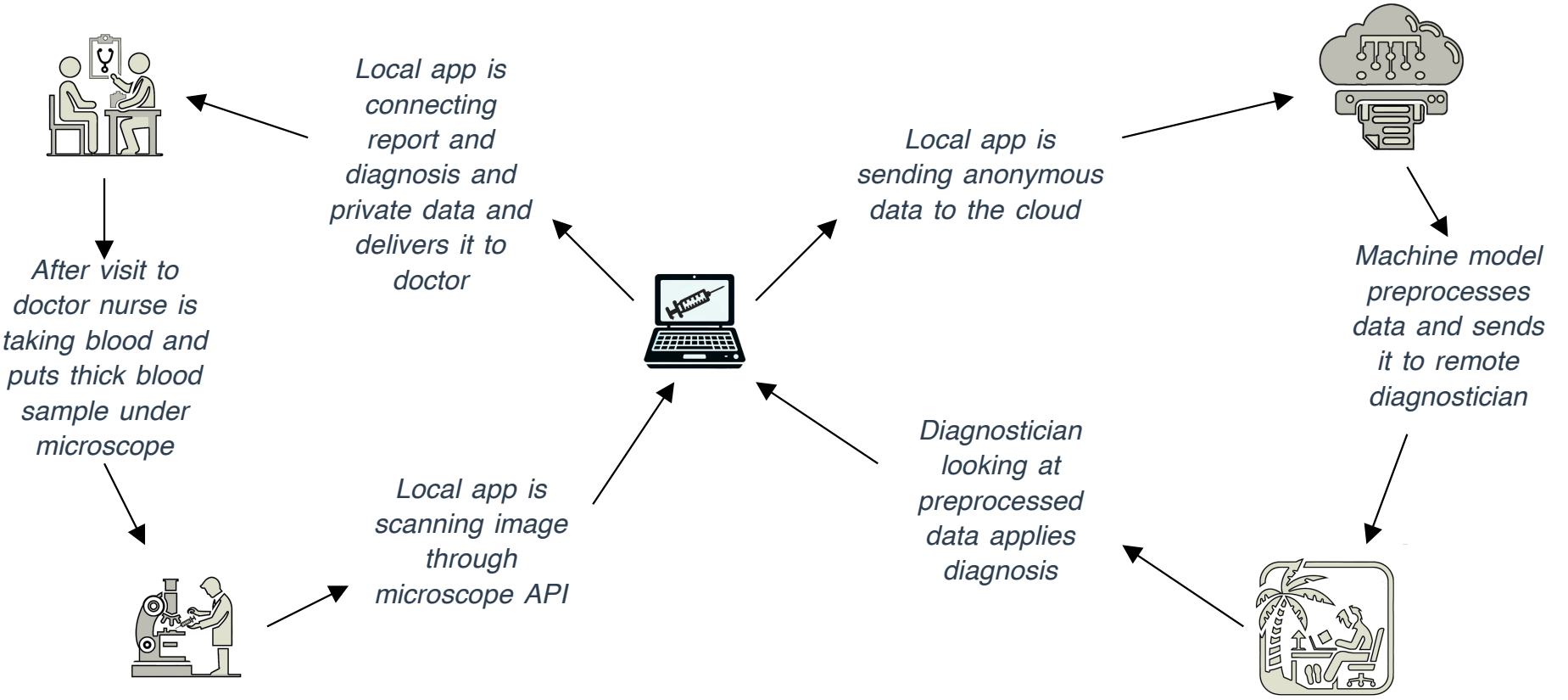
Gathered insights

- **Computer Vision is really good at detecting malaria parasites**
- Task of detecting presence of malaria requires extreme accuracy – we need to work with thick blood smears to avoid this limit
- Subject is super sensitive as it's about human life, to sensitive for humans to accept mistakes by computer - I we are not there yet, and approaching subject in one step is way too risky

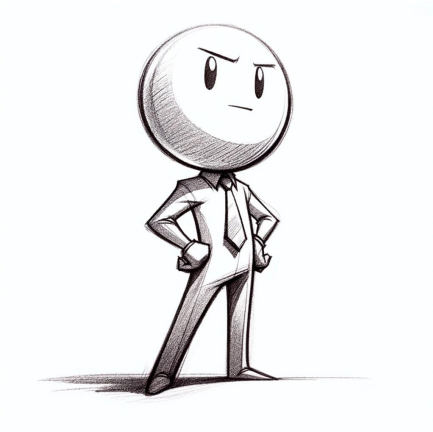
Refined plan

- **Let the software filter out all data with no any hints of malaria parasites**
- Return prepossessed image that is highly focused on suspected parts of image
- Reduce amount of data that humans must process by factor of ten
- Frees humans from mundane tasks, such as looking carefully at hundreds and hundreds of healthy cells that are of no interest
- Increase accuracy of diagnosis, as it helps keep human attention where it's required
- Increase throughput, as analyzing one tenth of data would take much less time

Solution – hybrid model concept



Next steps



- **review concept** with healthcare specialists

- **prepare cost analysis** – how much can it would cost per diagnosis and how that compares to existing solutions

- **prepare list of requirements for developing the product** so project wise cost analysis can be done