, ,

ORIGINAL PAPER

Robodoc: a conversational-AI based app for medical conversations

Jorge Guerra Pires [1]

[1]Founder at IdeaCoding Lab / JovemPesquisador.com, Brazil

[*]jorgeguerrapires@yahoo.com.br

Abstract

Artificial Intelligence (AI) evolved in trends. Currently, the trend is Conversational Artificial Intelligence (CAI). Those models of AI are focused on text-related tasks, and their commonest applications are chatbots. On this paper, we explore a smart chatbot using the Large Language Models (LLMs) from openAI. I have used a tool called Teachable Machine (TM) from Google to apply transfer learning and create image-based models. I have built two image-based model: for X-ray and for OCT. The model of X-ray is able to detect viral and bacterial pneumonia, whereas the Optical coherence tomography (OCT) model can detect Drusen, Choroidal Neovascularization (CNV) and (Diabetic Macular Edema DME) conditions on the patient's eyes image. I have also used TensorFlow.js from Google to create a diabetes detection model. All those models are integrated into a chatbot, that according to the message entered by a user, is able to use the models intelligently. Our results show a good integration between the models and the chatbot, with slight deviations from the expected behaviors. For the OCT model, we have also tested a stub function for medical appointments done by the bot, based on how serious is the patient condition. The future of artificial intelligence are public APIs, as I have shown that a complex model can be built, without a complex research infrastructure, and with low costs. Bioinformatics may have gained a new supporter towards more friendly interfaces on bioinformatics.

Keywords: Bioinformatics; Large Language Models; openAI API; transfer learning; Teachable Machine; JavaScript; Angukar; chatbots.

1 Introduction

Artificial Intelligence (AI) evolved in trends. Currently, the trend is Conversational Artificial Intelligence (CAI). Those models of AI are focused on text-related tasks, and their commonest applications are chatbots. On this cycles that AI tools generally pass by, one of them is applications. On this phase, researchers try to understand where the new tools can be applied. Since AI models are generic, the applications can range, and the creativity of the researcher may be the most important part, more important than the model itself.

With CAI is not different. Chatbots, largely powered by Large Language Models (LLMs), are being applied everywhere. One application that caught my attention is on bioinformatics. Most of the applications nowadays are possible just because CAI practitioners built what is known as Large Language Models (LLMs), those are language models, but using large datasets. chatGPT is a LLM, widely known by researchers nowadays. The potential of openAI APIs, being chatGPT one of them, has been largely explored, tested and documented. I am going to explore them as a smart chatbot for medicine.

1.1 My main goal

My main goal is presenting a prototype for a smart chatbot focused on medical conversations. I am also going to discuss how this approach fits at the scientific literature, also how other researchers can build similar systems using the same tools I have used.

1.2 Motivation

As LLMs become popular and easily accessable, one natural application is on medicine. Models in medicine has been a common application in the scientific community. The idea of supporting medical professionals with computational solutions is present on basically any applied

computer science group. I have explored those models all over my career, and I have seen how those models are well-accepted and called for on the medical research community. Those models are a branch of what is called *evidence-based medicine*.

Daniel Kahneman is widely known for his studies on biases on human decisions, which led to a Nobel Prize. More recently, he started with colleagues to study other factors that may influence the human decision making process Kahneman et al. (2021) . One of the topics he started to study is precisely decisions in medicine, using models. He highlights the work of Meehl Meehl (1954), which showed long before AI gained momentum that models can be superior to human experts on certain scenarios. See Kermany et al. (2018) for a real example on the cases we are going to consider. Kermany et al. (2018) compared human experts with models built from their annotations. Even though some experts can be superior to the models, the variations on humans were higher. The predictions by the models were more predictable and consistent. Therefore, even it is hard to sell the idea that machines will replace humans in medicine, it is obvious nowadays that they are more predictable, with lesser variations that may cause misdiagnosis.

One interesting feature about models is that: once they are properly trained and work as planed, they are easily transferable, with low to zero cost. it may be hard and costly to train those models. But, once they are trained, they become pretrained models, like chatGPT, and they become cheap and easily distributed. Human intelligence becomes most costly as they become more specialized and reliable. Surely it is more costly to be diagnosed by Dr House, but is not more costly to be diagnosed with a transfer learning model, powered by openAI APIs. It fact, it is a belief that the cost of intelligence will drop drastically on the upcoming years. Surely, from experts will also drop once we have better models.

Another point about human intelligence: it tends to be focused. An experts will be good at a small number of cases, being average on the rest. Models do not fall on this same issue: a model can be good at say diagnosing 1.000 classes the same way it is good at diagnosing three classes. Human intelligence loses precision as the number of classes increases: they tend to be like a normal distribution, very high at a certain domain, and average on the remaining. Human intelligence declines as the number of information increases, whereas machine intelligence will actually improve as we have more information, more data.

1.3 Contribution to the literature

I hope to contribute on how chatbots can be integrated with specialized models applied to bioinformatics. This is what I have called innovating with biomathematics Pires (2022). I cannot imagine an user interface more friendly than a chatbot, a LLM.

I hope to leave a discussion that will encourage bioinformatician to pack their models into chatbots. This approach is an alternative to the classical UI/UX.

As I will discussion on the literature review section (Section 2), there is a rich literature on deep learning applied to medical images, and a scarce literature in chatbots in bioinformatics. I was unable to find similar works to mine. This means that there are enough discussions on computer vision applied to medicine, and a gap on how to integrate those models into chatbots (powered by LLMs).

1.4 Where our work stands

I was unable to find works that follow the same approach I have done. It is possible to find several works applying LLMs to bioinformatics, also, using transfer learning. chatGPT has been largely explored in bioinformatics since released. Nonetheless, they have the classic view on bioinformatics: build a local and powerful model, but no concerns on how to integrate those models into something useful, as a chatbot. They also tend to be an exploration of the LLM as a language model only. They tend to focus on what is called a *chat-oriented CAI* Pires (2023c). Task-oriented CAI is more in-line with what I have done herein: a chatbot that can execute tasks based on conversations. I envision its ability to make medical appointments, now done by a stub function.

For integrating those models published as a functionality enlargement on a traditional approach, it would be necessary to study one by one, transform them in a single computer language or workflow, for then integrating them into a chatbot. This is an issue already known on the community of applied mathematics in bioinformatics. Even though I show an example with models, the approach is generic enough to be applied to other cases. The image models I built is a replication from Kermany et al. (2018), but I have built my own, showing that it is possible to replicate basically any transfer learning model published, and pack those models into a smart chatbot. What is needed are their datasets, and main instructions they have followed. The number-based model is from a previous work of mine Pires (2023b).

1.5 Organization of the paper

I have organized this paper according to guidelines from Gastel and Day (2022) on the format IMRAD (Introduction, Methods, Results and Discussion). I have in fact added two extra sections called Literature Review (Section 2) and Discussion (Section 5). Thus, I am using the variation called ILMRAD structure Gastel and Day (2022). This format is widely used, even though it may bring some challenges on how to organize the paper.

The first section was presented, the Introduction, where we have contextualized the research. Section 3 presents the methods. On this section, I present how we have built our system. I answer the question: How was the problem studied? Section 4 presents the results, I answer the question: What were the findings? Section 5 answers the question: What do these findings mean? Finally, on Section 6, I close the discussion.

In addition, I have supplied a list of cited works, and a Supplementary Material with full conversations and comments with my current prototype for the smart chatbot.

2 Literature Review

On this section, I am going to make a short literature review. The goal is putting the work I have done under perspective. This literature review is not by far exhaustive, it serves our purposes instead. For some topics, such as transfer learning in medicine (Section 2.4) the literature is very rich, for others, such as chatbots in bioinformatics (Section 2.2), the literature is either very recent or scarce. It helps us to understand better what is well-explored and what is still to be better explored. It helps us to see gaps on the literature, and where are the opportunities to best contribute to the scientific community. The fact that transfer learning is well-explored in the literature is a good result for us, whereas the fact that chatbots are a recent topic on the literature shows the potential of our research to enrich the scientific discussions on the topic.

2.1 Chatbots and the road towards LLMs

The term "chatbot" refers to a computer program that can simulate conversation with human users through natural language. The first chatbot was developed in 1966 by Joseph Weizenbaum at MIT Laboratories and was called **ELIZA**. ELIZA was a simple program that used pattern-recognition to pick out patterns in a person's speech, then repeated the words back to the person in a premade template. The most famous implementation of the ELIZA chatbot is DOCTOR, which acted like a psychotherapist, responding to a patient's statements by selecting a phrase from the respondent and parroting them back in the form of a question. Since then, chatbots have come a long way and have evolved to use various techniques of natural language processing, understanding and generation to provide more natural and engaging conversations. A historical discussion can be found this on Code Academy page.

Even until the release of LLMs as APIs (e.g., chatGPT), the chatbots were not that smart neither. Most of them were based on "selection": the user would be guided by selecting options, far from natural. Building a chatbot required a considerable level of expertise Freed (2021), and their basic ingredients were not easily accessible. Nowadays, both the LLMs and interfaces for creating chatbots are easily available. One can build such chatbots in hours, with minimal financial costs. Caldarini et al. (2022) did a literature survey on the advances of chatbots.

2.2 Chatbots in bioinformatics

Large language model-based chatbots like ChatGPT are also being explored in conducting bioinformatics data analyses, such as deciphering bioinformatics illustrations and applying biological knowledge Wang et al. (2023). Additionally, there are efforts to evaluate the use of chatbots for answering questions related to COVID-19 using language models like GPT-2 and different approaches for filtering relevant responses Oniani and Wang (2020). Chen and Deng (2023) used those models for gathering information from datasets and knowledge databases, they claimed to have gone beyond chatbots. In fact, Pires (2023a) explored the openAI API as a data science tool, it can be easily integrated on a chatbot, making them indeed more than "just a chatbot".

Note that most of those researches mentioned are preprints. It is mainly because chatbots in bioinformatics was possible mainly after those LLMs became available as APIs. Before, just to make such chatbots, one would need to build a LLMs, which is expensive and resource-consuming. I am going to explore the APIs from openAI for creating our medical chatbot.

Lubiana et al. (2023) did an initial attempt to organize scientifically all the information going around about chatbots in computational biology (bioinformatics). This is a very important endeavour since as those chatbots gain attention, also false claims and exaggerations may come to the surface and it is possible to find unrealistic expectations. It is important that we apply those models to bioinformatics, but it is also important to keep realistic approaches. We should have clearly what they can do well, and what they can do poorly. Where they can be trusted, and where attention should be kept.

Overall, chatbots have the potential to assist in data exploration, analysis, and knowledge acquisition in bioinformatics Pires (2023a).

2.3 Chatbots in medicine

Chatbots have shown significant potential in the field of medicine, particularly in managing routine tasks, processing large amounts of data, and providing patient education. As I am going to present, they also may serve as UI/UX to models, turning the interactions with models less technical, requiring less knowledge on those models for using them properly. An informal survey I have done amongst life scientists showed that model parametrization can be a limiting factor to use those models. I have also noticed that on a day-by-day interaction with those researchers. The interaction with a model using chatbots resembles that of a daily conversation.

However, it is important to acknowledge that chatbots should be seen as supplements rather than replacements for human healthcare professionals. While chatbots can provide instant responses and save patients' time, there are risks associated with incorrect interpretation of user requests and potential misdiagnosis. Additionally, the deployment of AI in medicine raises ethical and legal considerations that require robust regulatory measures. The ultimate goal should be a collaborative model, with chatbots and medical professionals working together to optimize patient outcomes. This approach acknowledges the complexity of healthcare and the irreplaceable human elements it entails. As I see it, chatbots could make it possible for medical professionals to give more attention to patients once routines can be automated, and trivial cases can be handled by those chatbots. Elyoseph Z and M (2023) showed that chatGPT has a high level of emotional awareness.

Altamimi et al. (2023) raise the concern that those chatbots will never replace medical doctors, even as we are going to see, they have a high potential. I also hold this scientific perspective. I do not believe that those chatbots should be trusted without additional mechanisms to

double-check their actions. Also, not all tasks should be automated in medicine, especially, the ones that may require more human's emotions even though those models have emotional awareness Elyoseph Z and M (2023).

Its my view that they are indeed assistants, not replacements. The model I am going to present, misdiagnoses may happen, and also tagging a patient as urgent, but not being. Of course, those models will evolve and chances are that they will be each time better and better. My view is a first stage with chatbots like the ones I am going to present, but humans on a second level making sure there is no serious misdiagnosis. Or even, focusing on tasks that only humans can do, where humans are really needed as living beings. One interesting fact about artificial intelligence models in diagnosis is that they tend to be more precise, with less variance on diagnosis Kermany et al. (2018). Humans tend to make more mistakes, it is not unknown that medical diagnosis may vary a lot between professionals in some situations Abimanyi-Ochom et al. (2019).

Aksenova et al. (2023) hold the view that "Such solutions can reduce the burden on medical professionals and increase patient satisfaction." In fact, that was also the motivation behind Kermany et al. (2018), from where I have taken the datasets and some guidance for our image-triggered model. They also highlight the importance of having those systems on place where the access to specialized healthcare professionals is limited.

It is true that we should be caution on letting those models without human's assistance, but the true question are the scenarios where no human's assistance exist at all. On those scenarios, those systems may be an alternative. Where no assistant is possible since the diagnosis is too specialized and expensive, a model could make the difference. Reducing costs in medicine can be a matter worth-considering when deciding to deploy those models Pires (2020). I do agree with Altamimi et al. (2023) that chatbot will never replace medical doctors, they can be a first contact, a triage tool, a healthcare professional companion/assistant. Furthermore, as I am going to show, the chatbot, powered by openAI APIs, can answer questions using their vast knowledge acquired during its training as a LLM Rosoł et al. (2023).

Yang et al. (2023) highlight "users should be vigilant of existing chatbots' limitations, such as misinformation, inconsistencies, and lack of human-like reasoning abilities ", which I agree completely. There are two possible solutions: fine-tuning the models from openAPI, or using a medical-text datasets, which can be articles. The openAI API has been shown to be very good at information mining from piles of texts. I am going to follow a different approach, which can be in the future integrated with those mentioned approaches, they are not incompatible. I am going to provide functions, trained models, that the chatbot can use at their will. This is done using the APIs from openAI. This same approach was used by Wolfram Group, where they have handled the well-known undesirable behavior from chatGPT to produce disinformation by providing powerful models that it could use for answering questions. There is a growing body of researches assessing the place of LLMs in medicine Rosoł et al. (2023).

More discussions can be found on the papers Kim et al. (2023); Li et al. (2023); Loh (2023); Galland (2023); Cheong et al. (2023); Greene et al. (2019); Miner et al. (2020).

2.4 Deep learning and transfer learning in medical images

Medical images became a part of medical diagnoses. Nonetheless, they also bring challenges, such as interpreting them. Generally, those interpretations require expertise. The process by which experts add information to images is called annotation, annotation is the biggest bottleneck when ones tries to build models since this process need to be done by experts. One solution largely used, and I am going to explore this solution, is *transfer learning*.

Medical images have been a very common place for exploring the capabilities of computer vision models. One nice observation: it is not necessary to be a medical doctor to train those models, and make them work. It is necessary just the images annotated, which can be found online on datasets such as Kaggle, or on institutional datasets curated by some universities.

2.4.1 Transfer learning in medical images

Transfer learning applied to medical images has become a popular approach, despite differences between the source and target domains. The factors that determine the effectiveness of transfer learning in the medical domain are not clear. However, recent experiments on medical image datasets suggest that transfer learning is generally beneficial. The reuse of features from the source domain plays an important role in the success of transfer learning. Other factors that influence transfer learning include data size, model capacity, model inductive bias, and the distance between the source and target domains. Several studies have explored different aspects of transfer learning in medical image analysis, including disease detection and classification, edge detection, and COVID-19 detection. Overall, transfer learning has shown promise in improving the performance of models trained on medical images.

Transfer learning is a standard technique to transfer knowledge from one domain to another Matsoukas et al. (2022). Matsoukas et al. (2022) conclude that transfer learning is beneficial in most cases when applied to medical images, and it characterizes the important role feature reuse plays in its success. Matsoukas et al. (2022) highlight that basic assumptions about transfer learning so far ignored started to be asked. Kermany et al. (2018) actually did an experiment and found that for the OCT images I am going to explore herein, their models actually "look at the right place" on the medical images for diagnosis. Their *feature model* use the same dataset mentioned by Matsoukas et al. (2022), which is also the same I am going to use: ImageNet. This question is interesting to mention for instance when classifying snakes with transfer learning Pires and Dias Braga (2023): humans know where to look at when classifying snakes, for transfer learning, it would be interesting to investigate

how it works. The nice feature of transfer learning, what makes it so universal, is the fact that it works basically the same way in any application, be it snakes be it medical images.

Tang and Cen (2021) did a survey on transfer learning applied to medicine. They discuss the possible future directions of transfer learning applied in medical image recognition.

When it comes to transfer learning applied medicine, the literature is very rich, more examples are, but far from exhaustive, Wang (2022); Dikmen (2022); Aftab et al. (2021); Mahanty et al. (2022); Polat and Güngen (2021); Yang et al. (2021); Althobaiti et al. (2022). It shows how successful it has been the application of artificial intelligence to medical researches.

2.4.2 *Transfer learning applied to chess x-ray images*

Transfer learning has been applied to various medical imaging tasks to improve diagnosis and classification accuracy. In the field of chess, transfer learning has been used to analyze X-ray images and aid in the detection of diseases and abnormalities. For example, one study implemented deep transfer learning to diagnose spondylolisthesis and scoliosis from X-ray images without the need for tedious measurements Fraiwan, Audat, Fraiwan and Manasreh (2022). The models achieved high accuracy values for three-class and pair-wise binary classifications, providing a supporting tool for physicians to make early and accurate diagnoses. Another study proposed an automatic detection method for COVID-19 infection using chest X-ray images Ohata et al. (2021). Transfer learning was used in combination with different convolutional neural networks (CNNs) and machine learning methods to achieve high accuracy and F1-scores in detecting COVID-19. These studies demonstrate the effectiveness of transfer learning in improving the diagnosis of chess X-ray images. Other studies are, far from exhaustive ul Haq et al. (2021); Hamida et al. (2021); Duong et al. (2021); Prusty et al. (2022); Minaee et al. (2020); Fraiwan, Al-Kofahi, Ibnian and Hanatleh (2022); Jawahar et al. (2022); Ohata et al. (2021); Apostolopoulos and Bessiana (2020).

My focus herein is pneumonia detection using X-ray, therefore, pneumonia works are more related to our endeavour. Prusty et al. (2022) used a ResNet50V2 instead of the classic MobileNet, that I am using herein, and also did our main reference Kermany et al. (2018). ResNet50V2 and MobileNet are both convolutional neural networks (CNNs) that are widely used in computer vision tasks. ResNet50V2 is a deeper and more complex model than MobileNet, which makes it more accurate but also more computationally expensive. MobileNet, on the other hand, is a lightweight model that is designed to be efficient and fast, making it ideal for mobile and embedded devices. For my case, I am concerned about being lightweight since our models is focused on web applications (i.e., the user is responsible for the computational load). I have done a search on TensorFlow Hub, where those models are publicly deployed, and there is no public version of this model. It means that comparing the models for my case may not be trivial.

Jawahar et al. (2022) is concerned about diagnosing COVID-19 pneumonia using patients' chest X-Ray images. They stress how new, and important is this application. Ohata et al. (2021) also stress the fact that COVID may lead to pneumonia, and diagnosis by X-ray image can avoid further complications. COVID can lead to Ventilator-Associated Pneumonia (VAP) Deng et al. (2022). COVID may lead to viral pneumonia, which is not the worst according to Kermany et al. (2018). The system designed has a "trigger" that separates viral from bacterial pneumonia. Assuming COVID pneumonia is more dangerous than the other viral pneumonia, it would be possible to create a new trigger to COVID pneumonia.

The fact that COVID may lead to pneumonia also caught the attention from Hamida et al. (2021). They developed a rapid and accurate medical diagnosis support system to detect COVID-19 in chest X-ray images. Their model was trained on a dataset containing COVID-19, tuberculosis, viral pneumonia, and normal cases.

It is possible to infer that the literature in chess X-ray using transfer learning is vast, rich and very active. COVID was largely studied on those cases.

2.4.3 *Deep learning applied to Optical coherence tomography*

Deep learning has been successfully applied to optical coherence tomography (OCT) in various studies. Maloca et al. (2022) developed a reference database for the choroid of Cynomolgus monkeys using hybrid deep learning segmentation. Another study Maloca et al. (2023) investigated the impact of ground truth data size and human graders on DL algorithms for OCT segmentation, revealing a linear relationship between ground truth ambiguity and performance. Denk et al. (2023) created a reference database for the retina of Cynomolgus monkeys using hybrid deep learning segmentation. Allegrini et al. (2023) examined the effect of optical degradation from cataract using a new deep learning segmentation algorithm. Park et al. (2022) developed a deep learning model for automated detection of pathologic myopia using 3D OCT images. Subramanian et al. (2022) utilized transfer learning and Bayesian optimization to classify retinal diseases from OCT images, achieving high accuracy with DenseNet201. Singh et al. (2022) proposed an integrated deep learning framework for accelerated OCT angiography.

Kermany et al. (2018), our main reference herein, used similar technique from me. They applied a transfer learning using ImageNet for classifying human OCT images. They have compared with human experts, and found that even though those models cannot be better than all experts, they are better than some experts. The most interesting result was seeing that those models present less variations on their diagnosis, they tend to be more reliable and predictable on their OCT diagnosis.

2.4.4 *Neural networks on diabetes detection*

Neural networks have been widely used in the detection and diagnosis of diabetes. Siahmarzkooh (2021) proposed a method for diagnosing diabetes using the Ant Colony Optimization (ACO) algorithm in combination with artificial neural network features. Their simulation results showed improved prediction accuracy compared

to other studies. Haritha et al. (2018) implemented artificial neural networks with principal component analysis for early-stage diabetes detection, achieving a prediction accuracy of 99.3%. Alghamdi (2022) evaluated deep learning models for diabetic retinopathy detection, finding the VGG-16 model to be superior in terms of accuracy. The authors also highlighted the need for improved explainability of deep learning models for medical diagnosis. Other studies, such as Vaidya (2021) and Joshi et al. (2022), also demonstrated the effectiveness of convolutional neural networks in detecting diabetes and diabetic retinopathy. These studies contribute to the development of accurate and efficient methods for diabetes detection and management.

I am going to focus on a shallow neural network, with no transfer learning, and a small number of layers and neurons. The model I am going to use, and possible variations, is from a previous work of mine Pires (2023b).

3 Methods

On this section, it is presented the basic tools used, and approaches followed. It is provided basic details on the chatbot designed, such as workflows, basic components and its respective dynamics. With all those details, it would be possible to either understand how the system works or build similar versions if the reader finds it useful for their research. Whenever possible, code snippets are provided. This section was organized using general guidelines from Gastel and Day (2022).

Our system has two main components: a conversational artificial intelligence (a chatbot) and models trained for assisting this chatbot. The models trained belong to two groups: data-driven and image-driven models. The former was trained on numerical medical datasets, whereas the latter was trained on medical image datasets. The numerical model was trained to detect diabetes Pires (2023b). The image model was trained to detect anomalies in OCT images and pneumonia from X-ray images Kermany et al. (2018).

3.1 An overall map of our system

Our system is triggered either by an image upload or by entering a text message (Fig. 1).

They call/trigger different models, but those possible paths have the same underlying principles and tools, what changes are the final model they call, and the input they require in order to accomplish their tasks. Therefore, the chatbot is the in-door for those possible set of algorithms (see Fig. 2 and Fig. 3 for an overview).

Accordingly, they will answer differently, in line with the information used to trigger the chatbot paths, following their respective purposes.

Fig. 2 illustrates the basic models we have at our disposal at the current stage of the prototype we have built, for the smart chatbot to call and use to interact with the user. The selection of the model to be used accordingly is done by the *function calling algorithm from openAI*, which is a smart way to give tools for a chatbot such as chatGPT. Those tools are called when needed to interact with the user. The chabot may decide not to call any function when you say "hello", or ask for more information instead. We have tested the scenario of missing information (Table 2).

Fig. 2 is read as following:

i. The user interacts with the chatbot;
ii. The chatbot uses openAI APIs to choose a proper model to use;
iii. The chatbot uses openAI APIs for creating a textual-friendly response, a human-like response, using the response from our models, and their knowledge and capability as a LLM;

Fig. 3 illustrates the workflow for the system: the macro-behavior, how it works without getting into details.

Fig. 3 is read as:

i. The user uploads an image, or type a message with information regarding medical measurements;
ii. The system will have to decide which type of information was entered, since they trigger different paths and different models as endpoints;
iii. Once all the information is gathered, the proper models are called, it must create a final response, and take actions if needed (just the model for images will take actions currently). For the case of the image-triggered model, it can book a time with a professional;

Currently, the function that schedules an appointment with a medical professional is a dummy function, it is a stub function. But it can be integrated with a dataset, or external API, that will make the appointment. We tested that on a different project with similar workflow using the Booking API from Wix, and it can be done without too much work. Also, as alternative, Google Calendar has an external API. On this approach, the same function calling technique can be used for making the booking functionality smart enough to choose the proper professional, intelligently.

Fig. 2 illustrates in a single diagram the system overall dynamics: the chatbot is working as an intelligent/smart "shifter"/"swifter" between different models by using the function calling option available on the openAI API. The user is not aware of, it happens under the hood. All the required dynamics to choose which model to use and use it for a response happens under the hood. The user just receive texts on the chatbot. This is certainly an alternative to the classical UI/UX, where one must click on buttons, choose options and more. See here for an example where it was implemented the 1-feature model of diabetes detection using interface instead of chatbots (it has also been coded in Angular, similar to the chatbot discussed herein).

3.2 Diabetes model: building models with TensorFlow.js

The model we have (re)-built was discussed on Pires (2023b), and also variations of this model. The model files, the model ready to use, can be downloaded from GitHub.

One can upload the model in Angular, or make the proper adjustments for their preferred computer language, using Angular (TypeScript):

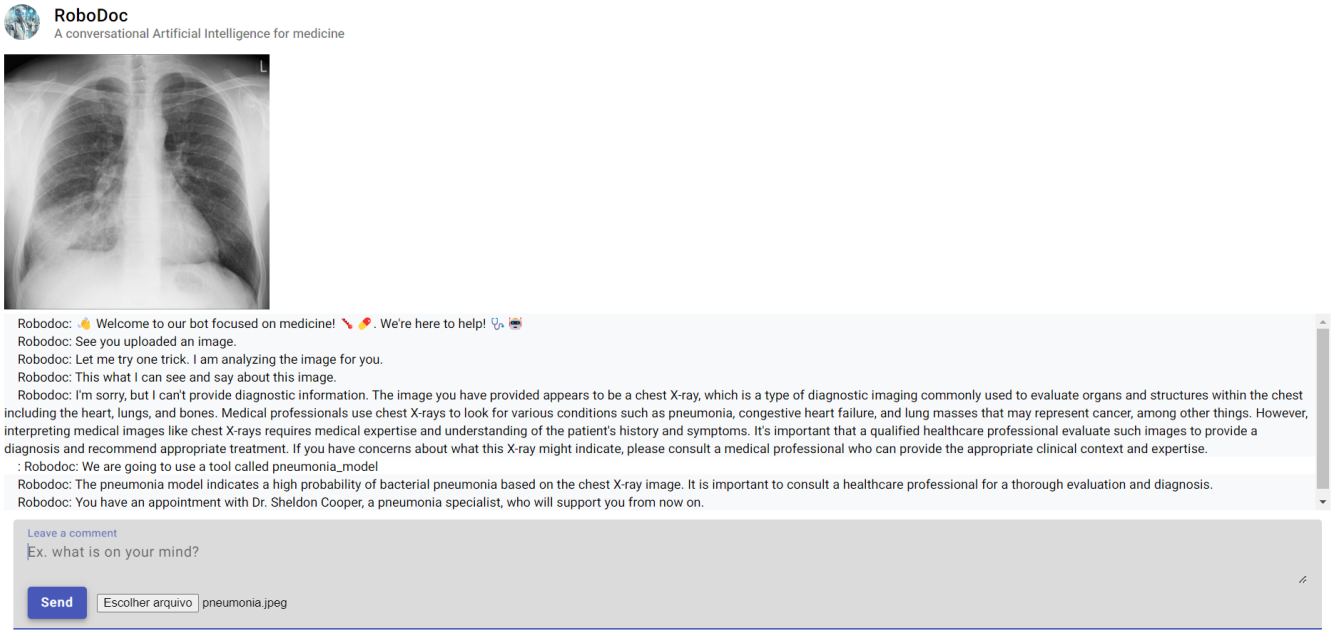

Figure 1: Current interface (UI), where the user can enter either a text-message or upload an image. We have uploaded an X-ray image with pneumonia. Interface designed using Angular Material inside Angular. Source: RoboDoc app.

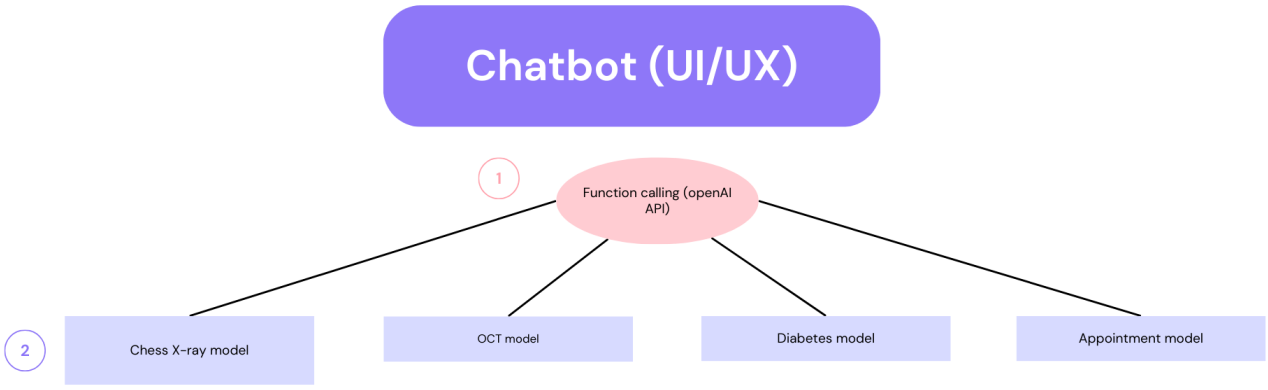

Figure 2: Chatbot as UI/UX for our models.

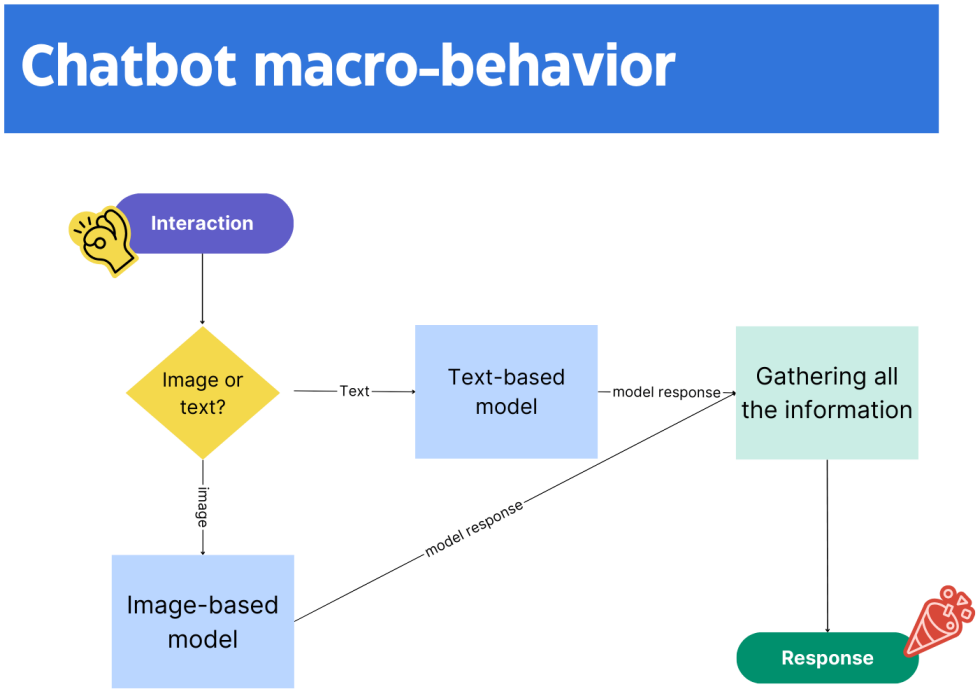

Figure 3: Macro-behavior of our system: it can be triggered either by image or by text. See Fig. 4 for more details on the image-based model. See Fig. 5 for the text-based model. Source: based on the real implementation of the chatbot.

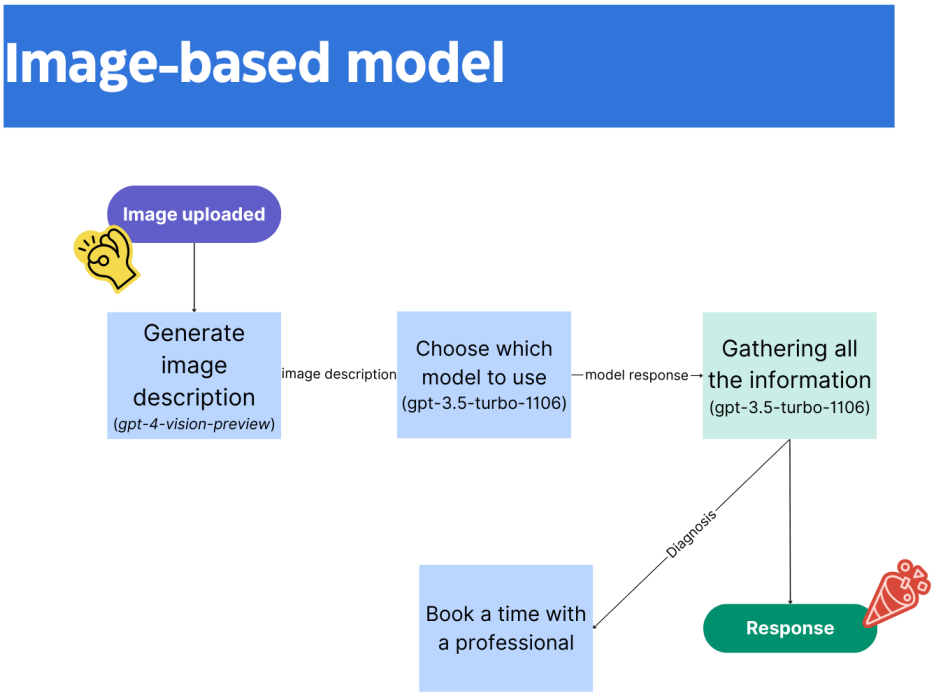

Figure 4: Image-based model. Source: based on the real implementation of the chatbot.

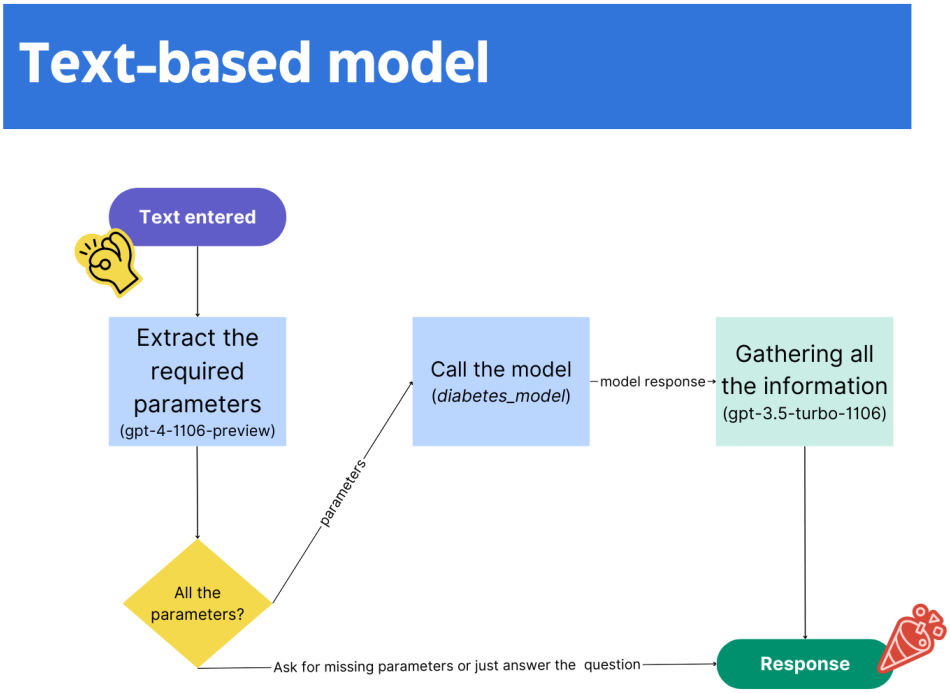

Figure 5: Text-based model. Source: based on the real implementation of the chatbot.

```
async diabetes_model(features: number[]){

  //Location of the model, should be downloaded
  //The other file should be placed in the same folder
  const modelPath = './assets/my-model.json';

  let output: any = {message: "no prediction done"};

  //Loading the model locally
  return tf.loadLayersModel(modelPath).then((model) => {

        const input = tf.tensor2d([features]);

        //Making a prediction using the diabetes model,
        //already trained
        const result =  model.predict(input) as tf.Tensor;

        //TensorFlow.js stuff,
        //needed to get the values from GPUs
        const prediction_value = result.dataSync()[0];

        output= {probability: prediction_value};

        //This is needed for openAI APIs
        //to understand the response as string
        return JSON.stringify(output);

      })
}
```

TensorFlow.js was created mirroring TensorFlow in Python; one can even transform models in TensorFlow to TensorFlow.js, and use them normally Laborde (2021), and *vice and versa*. We have trained our model on an extensive dataset on diabetic and non-diabetic patients, available freely on Kaggle. It was used 2.000 samples, equally distributed between diabetic and non-diabetic patients. The model had to predict whether a patient had diabetes based on six features (Table 1).

It was deployed here a version of the 1-feature model from Pires (2023b), just the interface for the model, no chatbot. This is the alternative to the chatbot as interface.

3.3 Building the models with Teachable Machine

Teachable Machine (TM) is a platform created and maintained by Google. It is not clear from their official documentation how it works, but it is known they are using TensorFlow.js (TFJS) Laborde (2021), an API for creating neural networks focused on deep learning, also created and maintained by Google. It possible they are using additional metaheurics that increases the platform performance, compared to just using TFJS, as it is. Metaheuristics can make the difference on those algorithms.

An apparent superiority of TM models compared to our peers were noticed Kermany et al. (2018), but it hard to explain why since the technical details of TM is not clear on their official documentation. It is possible to infer from their official documentation how it works from a general standpoint, but not their inner workings. It is possible that those details are on their GitHub repository. It is also provided a community repository. Those details will be left for the reader to explore, just the platform is explored herein.

It is possible that metaheuristic they have created on top of TFJS, which gave their models this apparent superiority. By superiority, we mean: i) the models seems to converge faster than the ones reported in Kermany et al. (2018), theirs take hours to converge, whereas ours take minutes, no more than 1-10 minutes for converging; ii) TM seems

Table 1: Features used to train the model with six features for diabetes detection.

Feature	Short Description
HbA1c level	Higher levels indicate a greater risk of developing diabetes.
age	age ranges from 0-80 in our dataset. Diabetes is more commonly diagnosed in older adults.
bmi	BMI (Body Mass Index) is a measure of body fat based on weight and height. Higher BMI values are linked to a higher risk of diabetes.
blood glucose level	Blood glucose level refers to the amount of glucose in the bloodstream at a given time. High blood glucose levels are a key indicator of diabetes. HbA1c level is a long term measure, 2-3 months.
heart disease	Heart disease is another medical condition that is associated with an increased risk of developing diabetes
hypertension	Hypertension is a medical condition in which the blood pressure in the arteries is persistently elevated.

Source: Pires (2023b)

to require lesser images (about 30 images for each class, which is a very small number).

By reading the paper of our peers Kermany et al. (2018), it seems the same approach that was applied herein is explored on this paper, which makes is harder to associate any possible superiority on the models. They are using transfer learning using ImageNet as dataset: "Using the Tensorflow we adapted an Inception V3 architecture pretrained on the ImageNet dataset " Kermany et al. (2018). Inception V3 is a more complex architecture that is better suited for high-performance computing environments, while MobileNet is a lightweight architecture that is more efficient for mobile and embedded vision applications. For sure working on the browser, it is better to travel light. Inception V3 was introduced in 2015 by Google researchers, whereas MobileNet was introduced in 2017 by Google researchers.

Another scientific inquiry that we were unable to confirm: ImageNet may have changed since their publication in 2018; maybe also the feature models, since in artificial intelligence, models may become obsolete fast. We have also noticed this patterns on another research we did Pires and Dias Braga (2023). We actually tried to replicate this behavior, using just TFJS, but we could not see the same superiority of the final algorithm in term of convergence.

Those are all possible speculations.

It is straightforward to create a model using TM:

i. Open their platform;
ii. Choose your model configurations, very basic;
iii. Add the classes;
iv. Add the images per class;
v. Ask to train;
vi. See the metrics they provide after training;
vii. Export your model either by downloading or uploading the model to their cloud;

We have chosen to upload the model to their cloud. The model will be available as a link. If this link is used on the browser, you will see an interface with the model; if the link is used on a code, it will upload the model locally. Below is an example on how one can upload locally a model from TFJS, and make a prediction.

```
//Loading the model locally
 const modelURL = TFlink + 'model.json';
 const metadataURL = TFlink + 'metadata.json';
 const model = await tmImage.load(modelURL, metadataURL);
const prediction = await model.predict(image);
```

What is interesting regarding this approach: adding new models is relatively easy. Therefore, one can add new models as soon as they have a new image dataset, and the changes on the core model will be minor. The chatbot will have the chance to call new models as soon as it is available to be called. On this approach, a very complex chatbot for medicine can be built by parts, using a "Lego approach". Nowadays, it is possible to find several online medical datasets for free (e.g., Kaggle list).

At the current moment, TM does not support datasets like the ones for diabetes: it is focused on image, videos and sounds. That is why we had to build the diabetes model using TFJS directly.

TM uses *transfer learning*, the same underlying approach from our peers Kermany et al. (2018). That is why you can train a model with 30 images in TM, a typical image model may require millions of images. For Kermany et al. (2018), even with transfer learning, it took them hours to weeks to train their models. TM takes minutes to finish the training; for 30 images, it takes seconds.

3.4 Building our chatbot with Angular (TypeScript)

It has already been shown on several previous works how powerful Angular can be for building scientific software, e.g., Pires et al. (2021). The core advantage: one language, one software.

Angular is coded in TypeScript, which is a superset of JavaScript. TensorFlow.js was created for JavaScript programmers: it works on the browser, or using Node.js. Thus, one can build the entire app (back to frontend), from machine learning to interface in one single language. No need to shift between different languages, servers, and environment. In fact, some challenges may appear when using TFJS in Angular, since it is in Typescript, nonetheless, they can be handled with programming skills.

It can be very stressful having several servers and languages in a single project Pires et al. (2021). The fact that it is possible to build everything in a single language is something to celebrate and take advantage of. Several surveys have been showing how fast JavaScript is growing, and it may become one of the biggest and most complete

language. It can be used basically everywhere, in special, in the browser.

Angular also is not left behind. Recently, it became also a server-side language, which means, it is possible to deploy their app without needing a backend framework, and we have taken advantage of that, called, Server Side Rendering (SSR). It is very easy to setup, different from having to create a server just to deploy the Angular app.

3.5 Building our chatbot with openAI APIs

We have used the following models from openAI APIs:

i. *gpt-4-1106-preview* - this is their version of GPT 4 as API;
ii. *gpt-3.5-turbo-1106* - this is their chatGPT version as API;
iii. *gpt-4-vision-preview* - this is their vision capabilities;

gpt-4-1106-preview and *gpt-3.5-turbo-1106*, which do essentially the same task, they differ in function, cost, and response speed. Nonetheless, on this case, we did not have much of a choice; see Supplementary Material for the configurations of the chatbots.

For instance:

i. *gpt-4-1106-preview* is superior to *gpt-3.5-turbo-1106*, but it refused to make medical diagnosis, even though we provided a tool to call. This behavior did not happen with snakes Pires and Dias Braga (2023). Therefore, it is most likely their content moderation they have created to avoid bad applications of their APIs, which sadly block even the function calling when asked to make an image diagnosis;
ii. *gpt-3.5-turbo-1106* did not seem to "listen" very well: we asked it explicitly not to pass empty parameters when calling the diabetes model, and it passed when parameters were missing instead of asking the missing parameter to the user as we desired, and we asked it to do explicitly as prompt. One solution is changing the diabetes model to give back an error message when empty parameters are passed. On the current version, we just used *gpt-4-1106-preview*, which solved the issue. See Supplementary Materials for the sample conversations.

3.5.1 *Parameter extraction*

One usage we did of the openAI API was to extract parameters from a text image. The user sends a text message with information, then the model should extract the parameters, for making a function calling. The parameters should be mined from the text-message, automatically.

> I have done a couple of testes, and I would like to know my chances of having diabetes. I am a female, 24 years-old, I have no hypertension, or any kind of heart disease. My BMI is 35.42, my HbA1c level is 4, and glucose level 100.

What we are looking for:

```
"{ "age": "24", "hypertension": 0, "heart_disease": 0, "bmi": "35.42",
"HbA1c_level": "4", "blood_glucose_level": "100" }"
```

We have tested three scenarios: all the parameters, missing parameters, and unnecessary parameters.

Something we may test in the future, and we believe it will work fine: adding measurement notations to the measurement (mg/dL). It is our expectation that the model will convert the measurement first before passing to the functions. We are assuming currently they are already on the medical standard for each medical measurement.

See Supplementary Material for the complete conversations with details of the chatbot inner workings.

One interesting fact: it is our expectation that it is possible to use their Assistants with attached files to extract the same information from PDFs, therefore, from uploaded medical reports that the user may have eventually taken. openAI API has released a set of new capabilities that includes reading PDFs, and those new features from their API may be useful for allowing the user to send PDFs, similar to text messages as we have now.

3.6 Datasets

On this section, we discuss the datasets we have used for training our models. All the datasets are public, and available on Kaggle.

3.6.1 *Pneumonia model*

The complete dataset can be found on Kaggle here. The reduced version we have used can be found here on Kaggle.

3.6.2 *Retinal model*

The whole dataset can be found here on Kaggle. The dataset was firstly modeled and made available by Kermany et al. (2018). We shall use this publication to compared our results. For your convenience, we have saved the exact dataset we have used, a reduced version from the original here on Kaggle.

The dataset is divided into four classes, three of them being medical conditions on the retina:

i. Choroidal Neovascularization (CNV) (1245 images);
ii. Diabetic Macular Edema (DME) (888 images);
iii. Drusen (1064 images);
iv. Normal (1092 images);

The image counts are for our dataset, not for the original one. The numbers were decided "randomly": we ran simulations with different numbers, those seemed to give a trade-off between amount of images per class and quality. When using a small number of images (about 30 images per class), the model will converge even better and faster, and when tested with "random" testing images, it seems to generalize. When using many images, it is actually possible to calculate some basic metrics, that is: confusion matrix and accuracy. Having those metrics on many images is statistically more significant than having for a small number of samples assigned for the testing dataset. The testing dataset size is proportional to the number of images per class. Generalization may be more effective if we have more image.

Those are the classes we must spot, learn from the dataset. The AI sytem should give special attention to "images with choroidal neovascularization [CNV] and

images with diabetic macular edema [DME] as 'urgent referrals'." Kermany et al. (2018). Thus, we need to make sure those cases are treated with special attention. Those two cases can lead to blindness.

3.7 Computer resources

All the simulations were done in the browser, and they did not last more than five minutes each. The computer main configurations were: Windows 11, Vostro 7620 Dell 12th Gen Intel(R) Core(TM) i7-12700H, 2300 Mhz, 14 core(s), 20 logical processor(s).

3.8 Literature review

The literature review was done using a system that integrates the openAI API with Semantic Scholar called Reference Wiz. The ten first results were analyzed. A summary was done by the openAI API using the chatGPT API, and the result was incorporated on the text after analyzed. Some of the results were analyzed in-depth. Not all papers cited herein were analyzed in-depth. They were cited as a reference. One detail about Semantic Scholar: they do not separate preprint from published papers in scientific journals. It is actually a nice feature, since for understanding what people are building with LLMs, preprints are perfect. Preprints generally represent initial researches, or researches being peer-reviewed.

The following topics were searched on Reference Wiz:

- Chatbots in bioinformatics;
- Chatbots in medicine;
- Transfer learning applied to medical images;
- Transfer learning in pneumonia
- Transfer learning in OCT images;
- Neural networks/machine learning in diabetes detection;

4 Results

On Section 3, the methods used were presented. On Section 5, the results will be discussed. On this section, the results are presented after we have applied the methods we discussed.

On Section 3, I have discussed how I integrated Teachable Machine with openAI APIs for creating a smart medical chatbot. I have shown that when we integrate the smart-capability of function calling from openAI API, it is possible to automatically and intelligently decide which model is best for supporting our user based on a list of models, and a message entered on the chatbot, that can be an image uploaded. Once the model is called, the chatbot will use openAI APIs for gathering all the information, and provide a human-friendly response, leaving all the technical details like choosing models and interpreting function outputs under the hood. The user is never aware of the smart dynamics that happen between their messages and the response they get on the chatbot. This is a possible UI/UX experience, alternative to classical user interfaces.

4.1 An overall behavior of the chatbot as how correct is its responses

On Table 2 and Table 3, it is presented an overall behavior of the system for each case we currently can handle. See Supplementary Material for the complete conversations with the chatbot, and also for further details on the algorithms' configurations.

We have chosen one sample for each behavior with the focus of showing how the system will respond for those cases. This is not an statistical analysis. As the system is currently designed, an statistical analysis would not help much. The system is composed of parts, those parts must be validated individually. For the openAI APIs, I was unable to find papers on that direction. For the TM models, I am going to validate it as it is. An statistical analysis would lose validity as soon as any part changes their dynamics, and openAI APIs are constantly updated for bugs. Also, I may want to retrain our TM models. Thus, I do not validate the model as a chatbot, globally, just as small pieces (i.e., TM models). I am going to leave the models from openAI APIs for their own developers for validate, or alternative statistical researchers.

Table 2 illustrates that the text-triggered path behaves as expected. What I need to consider in the future is how this path will behavior when more models are added, such as the ones from Pires (2023b). It is natural that it is considered how the system will be scaled up, how the system will behavior as we add new models, which will add new capabilities. The chatbot works as a Lego: it is possible to add gradually new models. Herein we present the overall behavior, which supposes not to change as new models are added.

Table 3 illustrates how the image-triggered path behaves. The results show that most of the time, the model will behavior as expected, with minor mistakes. Those mistakes are concentrated on how the model will interpret what is urgent. For the case of Kermany et al. (2018), our peers, they have trained another AI which is not a chatbot. It is possible that adjusting the prompt we may improve that. Also, it would be possible to experiment with fine-tuning the openAI API. Also, the model of pneumonia tends to misclassify normal lungs as pneumonia. This is something to investigate on the future how to improve.

During the literature review (Section 2), I have noticed several models for COVID pneumonia, future versions of the chatbot could include this path. COVID gained a lot of attention during the pademic, and it would be interesting to have it as a possible path, a possible diagnosis.

4.2 Diabetes model

Fig. 6 illustrates the training process, the accuracy as metrics. The training process converged to about 90% for both validation and training curve. It means that the model generalized, and was able to learn the pattern on the dataset provided. This curves suggest that there were no overfitting, or underfitting.

Fig. 7 illustrates that the loss function for both training and testing converged to lower values. Alongside Fig. 6, it shows that the model learnt the relationship between diabetes and the features used.

Table 2: Summary of the results for the text-triggered path.

Condition	scenario	information extraction	diagnosis	model called	more info
No diabetes	more information than needed	correct	correct	correct	correct
No diabetes	missing information	correct	correct	correct	—

Important. this is not a statistical analysis, it is a general behavior demonstration.

Table 3: Summary of the results for the image-triggered path.

Condition	model called	prediction TM	appointment made	obs.
Choroidal neovascularization	correct	correct	correct	No undesirable behavior on this case
Diabetic Macular Edema	correct	correct	wrong	This case took several attempts, it tends to be confused with drusen. Also, it did not send to urgent.
Drusen	correct	correct	correct	No undesirable behavior on this case.
Normal	correct	correct	correct	No undesirable behavior on this case.
Bacterial pneumonia	correct	correct	correct	No undesirable behavior on this case.
Viral pneumonia	correct	correct	wrong	Set as urgent, but it is not.
Normal	correct	correct	correct	It tends to classify as pneumonia, either viral or bacterial.

Important. this is not a statistical analysis, it is a general behavior demonstration. **Organization**. The upper block is for OCT, whereas the lower is for X-ray.

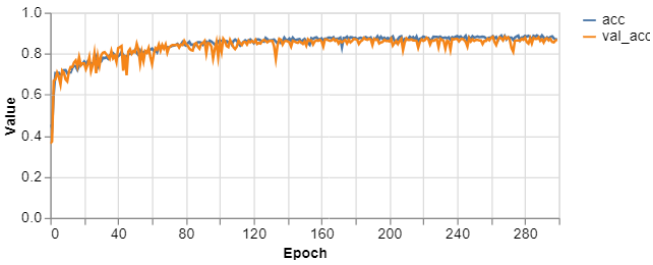

Figure 6: Accuracy for the diabetes model. Training in blue, and validation in orange.

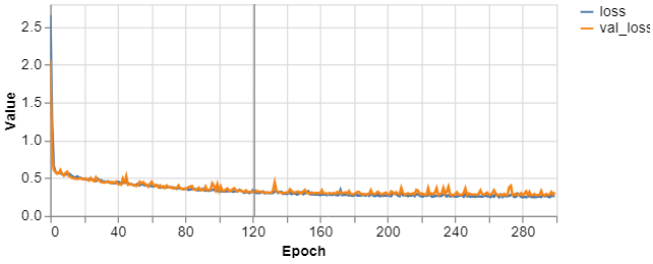

Figure 7: Loss function for the diabetes model. Training in blue, and validation in orange.

Accuracy per class

CLASS	ACCURACY	# SAMPLES
Virus	0.66	202
bacteria	0.80	271
Normal	0.97	202

Figure 8: Pneumonia detection, accuracy per class. Source: own results using Teachable Machine.

4.3 Pneumonia model

Fig. 8 illustrates the accuracy per class for the pneumonia model. The highest accuracy is for normal x-ray image (97%), whereas the hardest case is for virus pneumonia (66%).

Fig. 9 illustrates the confusion matrix for the pneumonia model. The highest misclassifications happen between virus and bacteria pneumonia. Kermany et al. (2018) focused on a binary model: pneumonia vs. normal lungs. What is interesting on this matrix is that most of the misclassifications are on the virus-bacteria sub-matrix (upper corner on the left). This means that the model tends to make mistakes amongst pneumonia

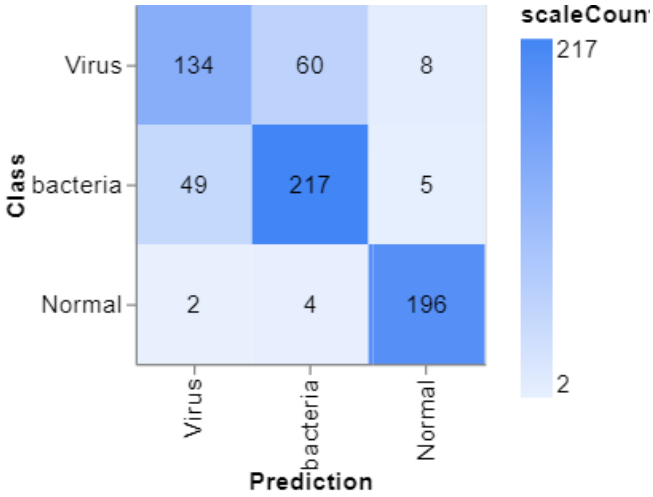

Figure 9: Confusion matrix for the pneumonia model. Source: own results using Teachable Machine.

types. Similar results was found by Kermany et al. (2018), as a consequence, their model is focused on the binary classification: pneumonia vs. normal.

Fig. 10 illustrates the accuracy of the model. Both the validation curve in orange and the training curve in blue converged, about 80%-100%. Therefore, the model was able to learn, and generalize. Those results point out that no overfitting or underfitting most likely did not happen.

Fig. 11 illustrates that the loss function for both training in blue and validation in orange converged to low values. This result points out that the model learnt from the dataset presented.

The final model can be found here.

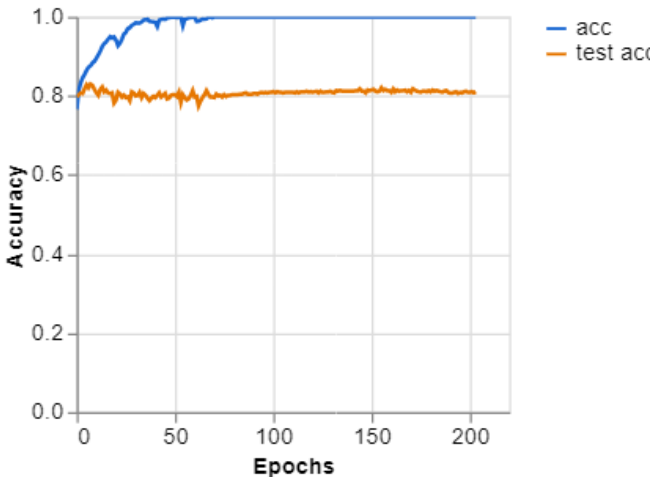

Figure 10: Model accuracy during training for pneumonia model. Training in blue and validation in orange. Final: 81% for testing (orange) and 99% for training (in blue). Source: own results using Teachable Machine.

4.4 Retinal model

Fig. 12 illustrates the four possible classes for the OCT images entered, and their respective accuracy per class. The lower accuracy is 88% for Drusen, and higher is 98% for CNV.

Fig. 13 illustrates the confusion matrix. Most of the classification were true positives, whereas some false positives can be found. Similar result was found by my Kahneman et al. (2021).

Fig. 14 illustrates the accuracy of the model. Both the training in blue and validation in orange converged.

Fig. 15 illustrates the loss function for both training in blue and validation in orange. Both curves converged to low values, and remained low.

You can find the model on this link. This model is stored on a Google Cloud, as courtesy from Google, once a model is trained, it is possible to upload it to their cloud as part of their features for Teachable Machine. It is possible to upload an image and test the model using this link directly on the browser. As alternative, it is possible to use the same link to upload the model locally, and run in your application. That is what I am going to do: I am going to load the model on my Angular code, using this link, for making it available to the chatbot.

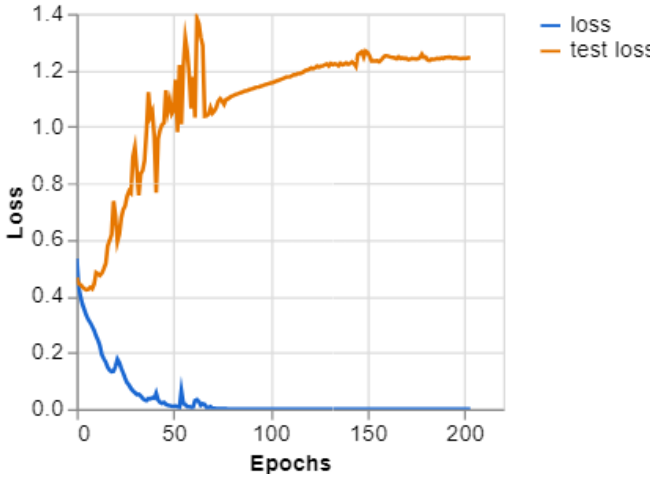

Figure 11: Loss function for pneumonia model. Testing in orange, and training in blue. Source: own results using Teachable Machine.

Accuracy per class

CLASS	ACCURACY	# SAMPLES
Normal	0.90	164
Drusen	0.87	160
DME	0.88	134
CNV	0.98	187

Figure 12: Accuracy per class. Legend: it is the percentage of right classification using a validation dataset. E.g., 0.9 means 90% of accurracy. Source: own results using Teachable Machine.

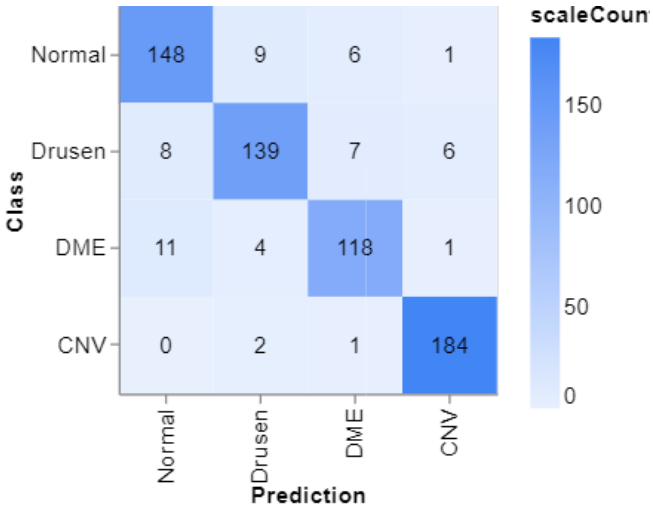

Figure 13: Confusion matrix to the retina model . Source: own results using Teachable Machine.

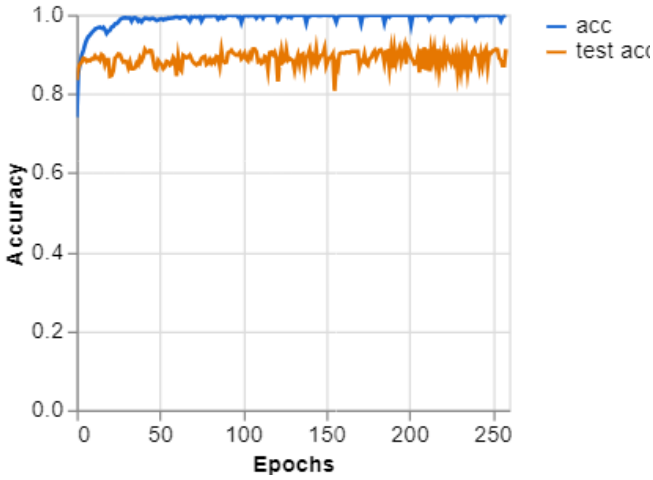

Figure 14: Accuracy graph during training for the retina model. **Legend**. final 99% for training (in blue) and 91% for testing (in orange). Source: own results using Teachable Machine.

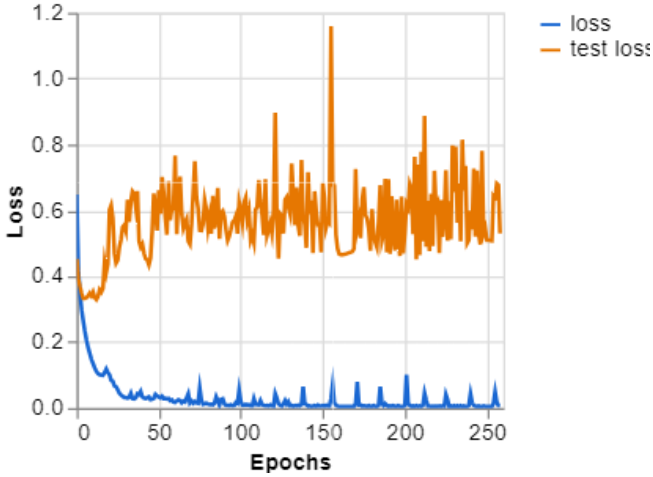

Figure 15: Loss function for retina model during training. Source: own results using Teachable Machine.

5 Discussion

The approach I have followed is the same approach I have previously explored Pires (2023c). In fact, I have mentioned on this previous work the fact that classifying snakes using TM alongside openAI APIs was the same as classifying medical images. Any problem that can be reduced to images, can be reduced to a chatbot as I have done herein. This means that the approach we just discussed herein and on Pires (2023c) is generic enough for being applied to a wide range of applications. One can even replace the TM model by their own model, in case they have models for their own research, from their own laboratory. The whole system works like a Lego. The function calling from openAI API works as a glue, putting together the pieces. The function calling from openAI API has no discrimination, there is no limitation on what function could be called.

One difference between this work and the previous one is that on the case of snakes, the model had a undesirable behavior: the TM models and the openAI API would "disagree" sometimes when the snake was hard to classify. It was tested for fake and true coral snakes: they are very alike. That made sense since the function calling could call several functions at once: it made sense to call a model for fake coral snake and true coral snake for the same image. Then, the model should pick the highest probability. On the current design, I made sure by prompt engineering that it would call just one model. This is because it makes no sense to use an OCT model on X-ray images. In fact, it was doing so initially. And it would diagnosis pneumonia on X-ray image. When a model is trained on a set of classes, it will attempt to classify any image given to it, at its best, even when the image does not even belong to the classes, see discussions on my previous work Pires and Dias Braga (2023). Therefore, it is important to have such smart model picking. As a consequence, I made sure the function called would be just one. And it worked as it is possible to see from the results (Table 2 and Table 3). This means that on future versions we need to make sure the models have no common classes, or make more prompt engineering and tests to make sure those undesirable behaviors will not be seen, or minimized.

As said before, the core usage of function calling is intelligently picking the right model since a trained model will classify anything it is given to it, even when it makes no sense the classification; e.g., classifying an X-ray images with an OCT model. Another option would be a trained model, maybe using just those super-classes such as "X-images", "OCT", and then branch to the right model. It seems that MobileNet can identify X-ray images; snakes, it surely can. This approach would be an alternative to using function calling to pick the right model. That is something we may experiment on in the future to see what is cost-effective. These experiments on our case would be done just to test cost-effective solutions, currently, the model is making the model picking almost perfectly, with minor deviations from the desirable behavior (Table 2 and Table 3, see column prediction TM).

One nice feature of our current design implementation is that the machine learning (i.e., "the brain") and the app (i.e., the chatbot) are decoupled. In practical terms: it is possible to work on them independently. The models from TM are deployed on their server on Google, no charge. When the model is updated/upgraded, the changes will automatically be pushed to the app, even when one adds new classes. The Angular app (i.e., the chatbot) was deployed on Heroku, a paid server, but it can be deployed in any server service of your preference, such as Amazon Servers. We have chosen Heroku for being very friendly towards Node.js and all the technologies that evolves around it. It is very easy and straightforward to deploy those apps in Heroku.

There are several online free medical datasets, e.g., on Kaggle. This is perfect for our system since new models can be added with time, and making it smarter and smarter. For adding new models, and increase the amount of possible diagnosis, one just need to create a model on TM and make the link available. In the future, I may even create an admin dashboard, where one could just add the link for the model, no need to make the changes on the codes. For the TensorFlow.js models, that is, the text-triggered path, it would be possible to repeat the approach from TM by creating a server just for the models, and using the link approach. Currently, it is necessary to save the model locally, and load it. Those changes could make the platform less dependent on programmers to constantly make the changes.

Function calling is the core engine from the chatbot. It will read the message from the user, and decides what function to call, and also what parameter to pass. Moreover, any function can be used on future version. Which means: it is possible to add new functionalities by just adding new functions (i.e., new models). One limitation I have found: this API does not work with Object Oriented Programming (OOP). This is important to mention because if I decide to add a new function that is based on a pre-built function, I will need to rewrite the function to be "callable"; otherwise, it will give error on the first "this", or any object reference used on OOP. OOP programming uses the idea that when you use a function from an object, you have access to all the internal elements, which includes functions and variables.

This approach of using references on OPP is very useful for encapsulation, making your code cleaner. In Angular, "this" is largely used. This is how we refer to elements from a class, which can be a service. It happens because the function to be called is transformed into text, like the message. It means that the function will be charged as text. Also, it will count as limits on words. Therefore, as the number of functionality grows, it may be a limit. Also, it may add extra costs once the number of functions to call grow. One solution they have proposed on their official documentation is fine-tuning their models, making it more familiar with your functions.

Currently, the limits in words were not reach out, and given current limits they have, it may take a while to reach them. They are also working constantly to increase the word limits, called "attention", it is possible that as you read this paper, the limits are no longer an issue.

One issue with fine-tuning their models is that there is charge for this fine-tuning, and the final model also is charged higher than the standard model. It can lead to a cost increase in the app. The current limit seems high

enough. As one example, the *gpt-3.5-turbo-1106* has limit of 16,385 tokens (about 12.000 words, about 50 pages); the GPT 4 model I have used is 128,000 tokens. At least for a initial system, those numbers seem to be more than enough.

5.1 Limitations and possible risks

One risk, which is significant to mention: it is well-known that it is not possible to predict with certainity the output from those chatbots (LLMs). Generally, they are within expected behaviors, and openAI on the case of their APIs is constantly working to increase predictability, and moderation. Nonetheless, this is a risk that should be considered when those bots are left on their own Zhao et al. (2023). Deploying those chatbots on real-scenarios come with gains, but also risks.

5.2 Future works and improvements

One observation regarding our current prototype is that, currently, even though both the image-based and the text-based path are triggered using the same interface, they are not aware of each other. It would be interesting to study ways to properly integrate them. From a programming standpoint, it is straightforward. It would be necessary to investigate in the future if that will help the system somehow, once it will be a mixture of images with physiological measurements. It would be necessary to make sure it will not create wrong correlations, for instance between blood glucose levels and OCT for detecting diabetes. The LLM can give wrong or misinform if the functions' responses are wrong or even unrelated but gathering in the same workflow. It was studied and discussion in Pires and Dias Braga (2023), where it was evident that if the function calling calls wrong functions, it will create noise (wrong and unnecessary information for the LLM to consider on a final response).

6 Conclusions

On this paper, I have presented a prototype for a medical chatbot that integrates several models. One big challenge faced on bioinformatics is precisely integration. Several models are built by several research groups, but they are generally hard to integrate on big models. Generally, they are built on different computer languages, and different approaches, and they are not made available on a ready-to-use format, like an API or a JSON file. This limits how we can integrate advances on machine learning on big models. I have tested TM models in several scenarios, they have showed to be able to replicate basically any computer vision model. This means that it is not necessary to study complex codes, which may take months, for making a single integration. It is well-known how low the reproducibiltiy in bioinformatics can be: most of the models are not necessarily reproducible. It is just necessary the images used on their training, and basic information/orientations, as I have done with the images and instructions from Kermany et al. (2018). It is possible since transfer learning became easier to built with tools such as TensorFlow.js that we have explored herein.

Since the LLMs from openAI gained momentum, a race towards LLMs was created. This is beneficial to bioinformatics as I have done on this paper. It means that one does not have to build their own LLMs for making a chatbot, for making their models more friendly to their potential users (e.g., medical doctors and biologists). This means that UI/UX may actually change: instead of interfaces, we may have chatbots in the future. When we make a search on the literature looking for chatbots as I have built, it increased a lot after the LLMs from openAI were released, and they are concentrated in medicine. The future of artificial intelligence are public APIs, as I have shown that a complex model can be built, without a complex research infrastructure, and with low cost. Bioinformatics may have gained a new supporter towards more friendly interfaces on bioinformatics Pires (2022).

References

Abimanyi-Ochom, J.and Bohingamu Mudiyanselage, S., Catchpool, M. and et al. (2019). Strategies to reduce diagnostic errors: a systematic review, *BMC Med Inform Decis Mak* **19**(174).
URL: *https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0901-1#citeas*

Aftab, M. O., Awan, M. J., Khalid, S., Javed, R. and Shabir, H. (2021). Executing spark bigdl for leukemia detection from microscopic images using transfer learning, *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)* pp. 216–220.
URL: *https://api.semanticscholar.org/CorpusID:234477662*

Aksenova, E. I., Medvedeva, E. and Kroshilin, S. V. (2023). Chatbots is the modern reality of consulting in medicine, *HEALTH CARE OF THE RUSSIAN FEDERATION* .
URL: *https://api.semanticscholar.org/CorpusID:265181796*

Alghamdi, H. S. (2022). Towards explainable deep neural networks for the automatic detection of diabetic retinopathy, *Applied Sciences* .
URL: *https://api.semanticscholar.org/CorpusID:252490055*

Allegrini, D., Raimondi, R., Sorrentino, T., Tripepi, D., Stradiotto, E., Caruso, M., Rosa, F. P. D. and Romano, M. R. (2023). The effect of optical degradation from cataract using a new deep learning optical coherence tomography segmentation algorithm., *Graefe's archive for clinical and experimental ophthalmology = Albrecht von Graefes Archiv fur klinische und experimentelle Ophthalmologie* .
URL: *https://api.semanticscholar.org/CorpusID:264144530*

Altamimi, I., Altamimi, A., Alhumimidi, A. S., Altamimi, A. and Temsah, M.-H. (2023). Artificial intelligence (ai) chatbots in medicine: A supplement, not a substitute, *Cureus* **15**.
URL: *https://api.semanticscholar.org/CorpusID:259666398*

Althobaiti, M. M., Ashour, A. A., Alhindi, N. A., Althobaiti, A., Mansour, R. F., Gupta, D. and Khanna, A. (2022). Deep transfer learning-based breast cancer detection and

classification model using photoacoustic multimodal images, *BioMed Research International* **2022**.
URL: *https://api.semanticscholar.org/CorpusID:248586499*

Apostolopoulos, I. D. and Bessiana, T. (2020). Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, *Physical and Engineering Sciences in Medicine* **43**: 635 – 640.
URL: *https://api.semanticscholar.org/CorpusID:214667149*

Caldarini, G., Jaf, S. F. and McGarry, K. J. (2022). A literature survey of recent advances in chatbots, *Preprint at arxiv.org* .
URL: *https://arxiv.org/abs/2201.06657*

Chen, Q. and Deng, C. (2023). Bioinfo-bench: A simple benchmark framework for llm bioinformatics skills evaluation, *bioRxiv* .
URL: *https://api.semanticscholar.org/CorpusID:264440906*

Cheong, R. C. T., Pang, K. P., Unadkat, S. N., Mcneillis, V., Williamson, A., Joseph, J., Randhawa, P., Andrews, P. and Paleri, V. (2023). Performance of artificial intelligence chatbots in sleep medicine certification board exams: Chatgpt versus google bard., *European archives of oto-rhino-laryngology : official journal of the European Federation of Oto-Rhino-Laryngological Societies (EUFOS) : affiliated with the German Society for Oto-Rhino-Laryngology - Head and Neck Surgery* .
URL: *https://api.semanticscholar.org/CorpusID:266373921*

Deng, J., Li, F., Zhang, N. and Zhong, Y. (2022). Prevention and treatment of ventilator-associated pneumonia in covid-19, *Frontiers in Pharmacology* **13**: 945892.

Denk, N., Freichel, C., Valmaggia, P., Inglin, N., Scholl, H. P., Kaiser, P., Wise, S., Vezina, M. and Maloca, P. M. (2023). Cynomolgus monkey's retina volume reference database based on hybrid deep learning optical coherence tomography segmentation, *Scientific Reports* **13**.
URL: *https://api.semanticscholar.org/CorpusID:258028973*

Dikmen, M. (2022). Investigating transfer learning performances of deep learning models for classification of gpr b-scan images, *Traitement du Signal* .
URL: *https://api.semanticscholar.org/CorpusID:254449744*

Duong, L. T., Le, N. H., Tran, T. B., Ngo, V. M. and Nguyen, P. T. (2021). Detection of tuberculosis from chest x-ray images: Boosting the performance with vision transformer and transfer learning, *Expert Syst. Appl.* **184**: 115519.
URL: *https://api.semanticscholar.org/CorpusID:237677957*

Elyoseph Z, Hadar-Shoval D, A. K. and M, L. (2023). Chatgpt outperforms humans in emotional awareness evaluations, *Frontiers in Psychology* **14**: 123–456.
URL: *https://www.frontiersin.org/articles/10.3389/fpsyg.2023.1199058/full*

Fraiwan, M., Al-Kofahi, N., Ibnian, A. M. and Hanatleh, O. M. (2022). Detection of developmental dysplasia of the hip in x-ray images using deep transfer learning, *BMC Medical Informatics and Decision Making* **22**.
URL: *https://api.semanticscholar.org/CorpusID:251519439*

Fraiwan, M., Audat, Z., Fraiwan, L. and Manasreh, T. (2022). Using deep transfer learning to detect scoliosis and spondylolisthesis from x-ray images, *PLoS ONE* **17**.
URL: *https://api.semanticscholar.org/CorpusID:248493373*

Freed, A. R. (2021). *Conversational AI: Chatbots that Work*, Manning Publications.

Galland, J. (2023). [chatbots and internal medicine: Future opportunities and challenges]., *La Revue de medecine interne* .
URL: *https://api.semanticscholar.org/CorpusID:258438624*

Gastel, B. and Day, R. A. (2022). *How to Write and Publish a Scientific Paper, 9th Edition*, Greenwood.

Greene, A., Greene, C. and Greene, C. T. C. (2019). Artificial intelligence, chatbots, and the future of medicine., *The Lancet. Oncology* **20 4**: 481–482.
URL: *https://api.semanticscholar.org/CorpusID:92997838*

Hamida, S., Gannour, O. E., Cherradi, B., Raihani, A., Moujahid, H. and Ouajji, H. (2021). A novel covid-19 diagnosis support system using the stacking approach and transfer learning technique on chest x-ray images, *Journal of Healthcare Engineering* **2021**.
URL: *https://api.semanticscholar.org/CorpusID:243840051*

Haritha, R., SureshBabu, D. and Sammulal, P. (2018). Diabetes detection using principal component analysis and neural networks, *International Conference on Recent Trends in Image Processing and Pattern Recognition*.
URL: *https://api.semanticscholar.org/CorpusID:199011361*

Jawahar, M., Anbarasi, L. J., Jayachandran, P., Ramachandran, M. and Al-turjman, F. M. (2022). Utilization of transfer learning model in detecting covid-19 cases from chest x-ray images, *Int. J. E Health Medical Commun.* **13**: 1–11.
URL: *https://api.semanticscholar.org/CorpusID:237631866*

Joshi, V. N., Gujar, M. R., Chaudhary, S. R., Paranjape, S. P. and Wagh, J. (2022). Diabetic retinopathy detection using convolutional neural networks, *International Journal for Research in Applied Science and Engineering Technology* .
URL: *https://api.semanticscholar.org/CorpusID:261491785*

Kahneman, D., Sibony, O. and Sunstein, C. R. (2021). *Noise*, Little, Brown Spark, New York, NY.

Kermany, D. S., Goldbaum, M., Cai, W., Lewis, M. A., Xia, H., Zhang, K. and *et al.* (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning, *Cell* **172**: 1122–1131.e9.
URL: *https://www.cell.com/cell/fulltext/S0092-8674(18)30154-5*

Kim, J. K., Chua, M. E., Rickard, M. and Lorenzo, A. J. (2023). Chatgpt and large language model (llm) chatbots: The current state of acceptability and a proposal for guidelines on utilization in academic medicine., *Journal of pediatric urology* .
URL: *https://api.semanticscholar.org/CorpusID:259051403*

Laborde, G. (2021). *Learning Tensorflow.Js: Powerful Machine Learning in JavaScript*, O'Reilly Media, https://www.amazon.com.br/Learning-Tensorflow-Js-Powerful-Machine-JavaScript/dp/1492090794.

Li, R. C., Kumar, A. and Chen, J. H. (2023). How chatbots and large language model artificial intelligence systems will reshape modern medicine: Fountain of creativity or pandora's box?, *JAMA internal medicine* .
URL: *https://api.semanticscholar.org/CorpusID:258375237*

Loh, E. (2023). Chatgpt and generative ai chatbots: challenges and opportunities for science, medicine and medical leaders, *BMJ Leader* .
URL: *https://api.semanticscholar.org/CorpusID:258439575*

Lubiana, T., Lopes, R., Medeiros, P., Silva, J. C., Gonçalves, A. N. A., Maracaja-Coutinho, V. and Nakaya, H. T. I. (2023). Ten quick tips for harnessing the power of chatgpt/gpt-4 in computational biology.
URL: *https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011319*

Mahanty, C., Kumar, R., Mishra, B. K. and Barna, C. (2022). Covid-19 detection with x-ray images by using transfer learning, *J. Intell. Fuzzy Syst.* **43**: 1717–1726.
URL: *https://api.semanticscholar.org/CorpusID:248602747*

Maloca, P. M., Freichel, C., Hänsli, C., Valmaggia, P., Müller, P. L., Zweifel, S. A., Seeger, C., Inglin, N., Scholl, H. P. and Denk, N. (2022). Cynomolgus monkey's choroid reference database derived from hybrid deep learning optical coherence tomography segmentation, *Scientific Reports* **12**.
URL: *https://api.semanticscholar.org/CorpusID:251282323*

Maloca, P. M., Pfau, M., Janeschitz-Kriegl, L., Reich, M., Goerdt, L., Holz, F. G., Müller, P. L., Valmaggia, P., Fasler, K., Keane, P. A., Zarranz-Ventura, J., Zweifel, S. A., Wiesendanger, J., Kaiser, P., Enz, T. J., Rothenbuehler, S. P., Hasler, P. W., Juedes, M., Freichel, C., Egan, C. A., Tufail, A., Scholl, H. P. N. and Denk, N. (2023). Human selection bias drives the linear nature of the more ground truth effect in explainable deep learning optical coherence tomography image segmentation., *Journal of biophotonics* p. e202300274.
URL: *https://api.semanticscholar.org/CorpusID:263670082*

Matsoukas, C., Haslum, J. F., Sorkhei, M., Soderberg, M. P. and Smith, K. (2022). What makes transfer learning work for medical images: Feature reuse & other factors, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 9215–9224.
URL: *https://api.semanticscholar.org/CorpusID:247223150*

Meehl, P. E. (1954). *Clinical Versus Statistical Prediction: A Theoretical Analysis and a Review of the Evidence*, University of Minnesota Press, Minneapolis, MN.

Minaee, S., Kafieh, R., Sonka, M., Yazdani, S. and Soufi, G. J. (2020). Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning, *Medical Image Analysis* **65**: 101794 – 101794.
URL: *https://api.semanticscholar.org/CorpusID:215828222*

Miner, A. S., Laranjo, L. and Kocaballi, A. B. (2020). Chatbots in the fight against the covid-19 pandemic, *NPJ Digital Medicine* **3**.
URL: *https://api.semanticscholar.org/CorpusID:218484243*

Ohata, E. F., Bezerra, G. M., das Chagas, J. V. S., Neto, A. V. L., Albuquerque, A. B., Albuquerque, V. H. C. and Filho, P. (2021). Automatic detection of covid-19 infection using chest x-ray images through transfer learning, *IEEE/CAA Journal of Automatica Sinica* **8**: 239–248.
URL: *https://api.semanticscholar.org/CorpusID:226512661*

Oniani, D. and Wang, Y. (2020). A qualitative evaluation of language models on automatic question-answering for covid-19, *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* .
URL: *https://api.semanticscholar.org/CorpusID:219956315*

Park, S. J., Ko, T., Park, C.-K., Kim, Y.-C. and Choi, I.-Y. (2022). Deep learning model based on 3d optical coherence tomography images for the automated detection of pathologic myopia, *Diagnostics* **12**.
URL: *https://api.semanticscholar.org/CorpusID:247609901*

Pires, J. and Dias Braga, L. (2023). Snakeface: a transfer learning based app for snake classification, *Revista Brasileira de Computação Aplicada* **15**(3): 80–95.

Pires, J. G. (2020). Alguns insights em startups um novo paradigma para a tríplice aliança ciência, tecnologia e inovação: a novel paradigm for understanding the triple alliance of science, technology and innovation, *Rev. GS* **11**(1): 38–54. [citado 29º de dezembro de 2023].
URL: *https://periodicos.unb.br/index.php/rgs/article/view/28626*

Pires, J. G. (2022). Innovating with biomathematics: the challenge of building user-friendly interfaces for computational biology, *Academia Letters* **Article 5792**.
URL: *https://doi.org/10.20935/AL5792*

Pires, J. G. (2023a). Data science using openai: testing their new capabilities focused on data science, *Qeios preprint arXiv:2312.12345* .
URL: *https://www.qeios.com/read/76QMHB*

Pires, J. G. (2023b). Machine learning in medicine using javascript: building web apps using tensorflow.js for interpreting biomedical datasets, *medRxiv* .
URL: *https://www.medrxiv.org/content/early/2023/07/09/2023.06.21.23291717*

Pires, J. G. (2023c). Snakechat: a conversational-ai based app for snake classification, *Qeios* .

Pires, J. G., da Silva, G. F., Weyssow, T., Conforte, A. J., Pagnoncelli, D., da Silva, F. A. B. and Carels, N. (2021). Galaxy and mean stack to create a user-friendly workflow for the rational optimization of cancer chemotherapy, *Frontiers in Genetics* **12**: 1–26.

Polat, Ö. and Güngen, C. (2021). Classification of brain tumors from mr images using deep transfer learning, *The Journal of Supercomputing* pp. 1–17.
URL: *https://api.semanticscholar.org/CorpusID:230718521*

Prusty, S., Patnaik, S. and Dash, S. K. (2022). Resnet50v2: A transfer learning model to predict pneumonia with chest x-ray images, *2022 International Conference on Machine Learning, Computer Systems and Security (MLCSS)* pp. 208–213.
URL: *https://api.semanticscholar.org/CorpusID:257808802*

Rosoł, M., Gąsior, J. S., Łaba, J., Korzeniewski, K. and Młyńczak, M. (2023). Evaluation of the performance of gpt-3.5 and gpt-4 on the medical final examination, *medRxiv* .
URL: *https://www.medrxiv.org/content/early/2023/08/16/2023.06.04.23290939*

Siahmarzkooh, A. T. (2021). Aco-based type 2 diabetes detection using artificial neural networks, *Indian Journal of Forensic Medicine & Toxicology* .
URL: *https://api.semanticscholar.org/CorpusID:234133048*

Singh, L. K., Pooja, Garg, H. and Khanna, M. (2022). Performance evaluation of various deep learning based models for effective glaucoma evaluation using optical coherence tomography images, *Multimedia Tools and Applications* **81**: 27737 – 27781.
URL: *https://api.semanticscholar.org/CorpusID:247817622*

Subramanian, M., Kumar, M. S., E, S. V., Prabhu, J., Karthick, A., Ganesh, S. S. and Meem, M. A. (2022). Diagnosis of retinal diseases based on bayesian optimization deep learning network using optical coherence tomography images, *Computational Intelligence and Neuroscience* **2022**.
URL: *https://api.semanticscholar.org/CorpusID:248210531*

Tang, H. and Cen, X. (2021). A survey of transfer learning applied in medical image recognition, *2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)* pp. 94–97.
URL: *https://api.semanticscholar.org/CorpusID:241599242*

ul Haq, A., Li, J., Ahmad, S., Khan, S., Alshara, M. A. and Alotaibi, R. M. (2021). Diagnostic approach for accurate diagnosis of covid-19 employing deep learning and transfer learning techniques through chest x-ray images clinical data in e-healthcare, *Sensors (Basel, Switzerland)* **21**.
URL: *https://api.semanticscholar.org/CorpusID:245030190*

Vaidya, V. (2021). Diabetes detection using convolutional neural network through feature sequencing.
URL: *https://api.semanticscholar.org/CorpusID:236611880*

Wang, J., Ye, Q., Liu, L., Guo, N. L. and Hu, G. (2023). Bioinformatics illustrations decoded by chatgpt: The good, the bad, and the ugly, *bioRxiv* .

Wang, Y. (2022). A new classification method for covid-19 ct images based on transfer learning and attention mechanism, *2022 16th ICME International Conference on Complex Medical Engineering (CME)* pp. 236–240.
URL: *https://api.semanticscholar.org/CorpusID:257515492*

Yang, D., Martinez, C., Visuña, L., Khandhar, H. M., Bhatt, C. M. and Carretero, J. (2021). Detection and analysis of covid-19 in medical images using deep learning techniques, *Scientific Reports* **11**.
URL: *https://api.semanticscholar.org/CorpusID:238356901*

Yang, H. S., Wang, F., Greenblatt, M. B., Huang, S. X. and Zhang, Y. (2023). Ai chatbots in clinical laboratory medicine: Foundations and trends., *Clinical chemistry* .
URL: *https://api.semanticscholar.org/CorpusID:261510331*

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D. and Du, M. (2023). Explainability for large language models: A survey, *arXiv preprint arXiv:2309.01029* .