Project Proposal: LuzTriagix

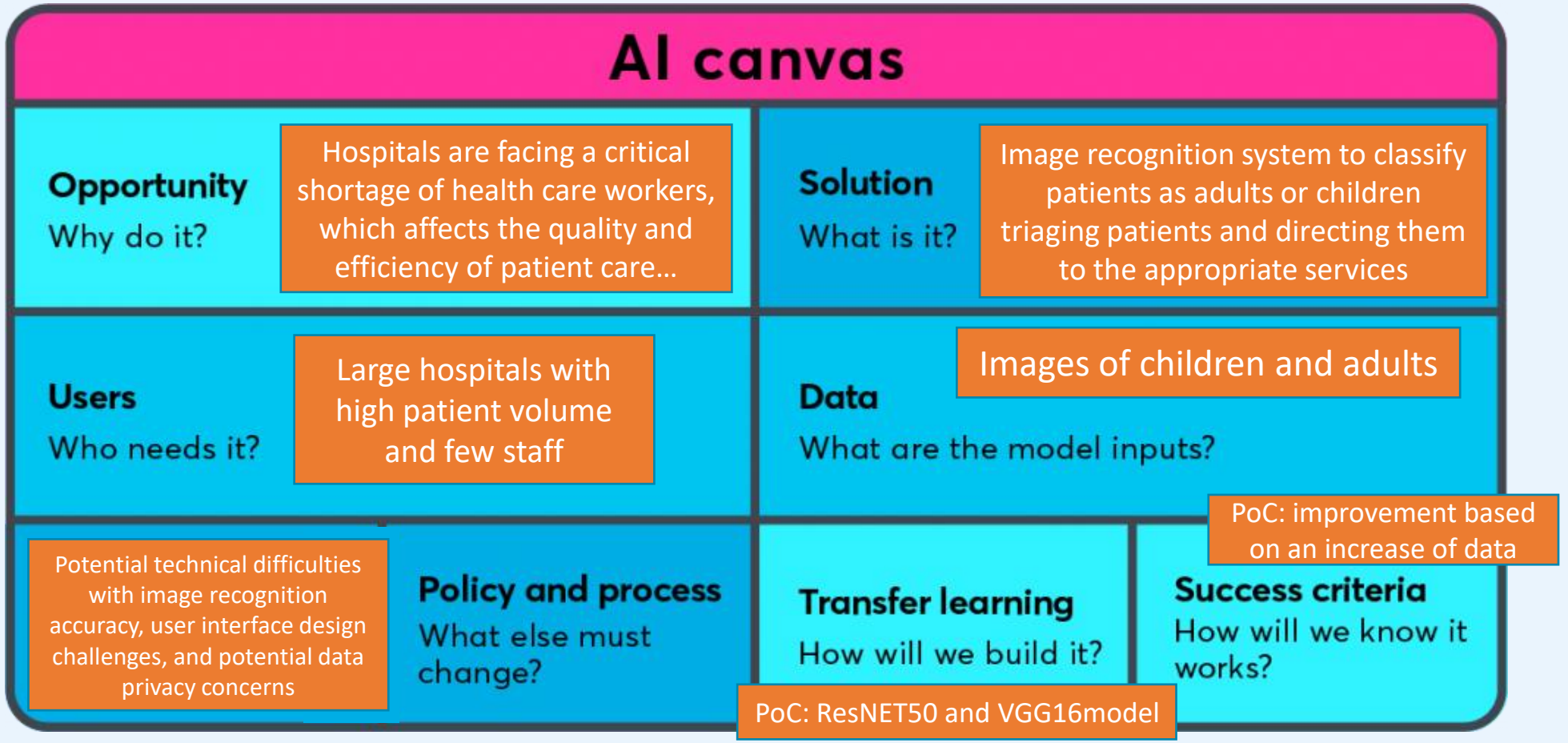
Francisco Mansilha
220387

Index

The presentation can be divided in the following segments

- AI Canvas – a high level overview of the project
- Market Research
- Business Proposal and User Persona
- Disruptive Technology Risks
- Responsible AI (Fairness and Bias)
- PoC: The Dataset and Preprocessing
- PoC: Baselines
- PoC: CNN Architecture and Training
- PoC: Model Performance and Error Analysis
- Responsible AI (Transparency & Interpretability)
- Human Centered AI - Application Design

AI Canvas: provide high-level overview



Market Research

The problem hospitals are facing

- According to the American Hospital Association, hospitals are facing a severe lack of healthcare workers, including nurses, especially during the Covid-19 pandemic. Since February 2020, hospital employment has decreased by nearly 94,000, including a decrease of over 8,000 between August 2021 and September 2021 alone.

- According to Forbes The Healthcare Industry Is Crumbling Due To Staffing Shortages.

- According to the World Health Organization, **one of the key strategies to address the health workforce crisis is to optimize the roles of existing health workers** through task shifting and delegation.

- According to the World Health Organization There is a global shortage of health workers, in particular nurses and midwives, who represent more than 50% of the current shortage in health workers.

Market Research

Are there triaging systems like this on the market?

- When searching for triaging systems on the health sector I found many results. However, these were not related to age classification but instead related to adult chest radiographs, mammography and others of sorts.

- Taking that into consideration I tried to search for triaging systems based on age and outside on a more global scope. One example I found was a tutorial by Adrian Rosebrock created on April 13 2020 in which he teaches how to perform age detection, prediction using OpenCV, Deep Learning and Python

Market Research

Target segmentation

- **Demographic segmentation:** Since hospitals or hospital administrators would be our clients, and they would not necessarily fall under the traditional definition of "customers." Therefore, demographic segmentation may not be applicable in this context
- **Geographic segmentation:** analyse geographically, which countries are more affected by the health worker crisis
- **Psychographic segmentation:** we could argue that we could try to segment hospitals based on the level of receptiveness towards the implementation of new technologies, but on the other hand one could say that, how much tech a hospital has is directed to its budget and not the receptiveness of the hospital administrator.
- **Behavioural segmentation:** Segment based on the hospital patient flow patterns

The most suitable target segment for this image classifier application is hospitals and healthcare facilities with a high volume of patients. Although it might also be important to look into which countries have a harsher healthcare worker crisis. Therefore, Behavioural and Geographical segmentation

Business Proposal

And problem statement

Problem statement

Hospitals are facing a critical shortage of healthcare workers, which affects the quality and efficiency of patient care. A CNN image classifier that can identify adults and children from images can help optimize the roles of existing staff by triaging patients and directing them to the appropriate services.

Proposal

Taking it into consideration, I propose an image classifier model that can recognize whether a person is an adult or child. This model can be applied in large-scale hospitals with the intent of optimizing healthcare workers' time and allowing them to focus on more important tasks than directing people. This image classifier model was designed to quickly and accurately direct patients to the appropriate departments and services within a healthcare facility. The system was trained on a dataset of people's faces, both children and adults of all ages and it was designed to be capable of co-learning and adaptation

The full business proposal with background, benefits, conclusion and references can be found in the documents folder

User Persona - Light Hospital

Background

Light hospital has been open for 10 years now. Throughout this decade they have acquired a long list of clients who trust the health care of this institute's professionals to provide them with care and treatments. This hospital and its administration is always open to innovation and dedicated to making the life of their patients better.

Hospital size and location

Large hospitals and healthcare facilities located in regions that have experienced healthcare worker shortages and emergencies.

Open since

2013

Patient flow

very high

"We're here for you, always dedicated and caring for you and your family

Goals

The hospital's primary goal is to provide excellent care to patients while reducing costs and improving efficiency. The hospital also aims to attract and retain top talent in the healthcare industry.

Pain points

The hospital has been struggling to provide quality and efficient patient care due to the shortage of healthcare workers

The hospital has been facing difficulties in managing patient flow due to the shortage of healthcare workers.

Disruptive Technology Risks

Risks

- Security risks: it may be vulnerable to hacking attempts or data breaches.
- Technical risks: it may experience technical issues that could cause it to crash or malfunction.
- Legal risks: it may not comply with relevant laws and regulations, which could result in legal action against the client.
- Privacy risks: it may collect sensitive user data, which could be misused or stolen.
- Ethical risks: it may be biased against certain groups of people, such as people of colour or biased towards people who appear significantly older/younger

Contingency Plan

- Security contingency plan: Implement security measures such as encryption and two-factor authentication to protect user data.
- Technical contingency plan: Regularly test the app for bugs and issues and have a team in place to quickly address any problems that arise.
- Legal contingency plan: Consult with legal experts to ensure that the app complies with all relevant laws and regulations.
- Privacy contingency plan: Implement privacy policies that clearly outline how user data will be collected, stored, and used.
- Ethical contingency plan: Regularly audit the app for bias and work to eliminate any instances of bias that are found.

Responsible AI (Fairness & Bias)

Bias, fairness, transparency and interpretability

Culture Bias:

When searching for “marriage” the majority of images depict only caucasian people and western weddings, while other types of weddings, such as Indian and Moroccan, are underrepresented.

Ramifications:

- It perpetuates cultural stereotypes and a narrow view of what a wedding should look like.
- It can limit the understanding and recognition of different cultural traditions and norms.

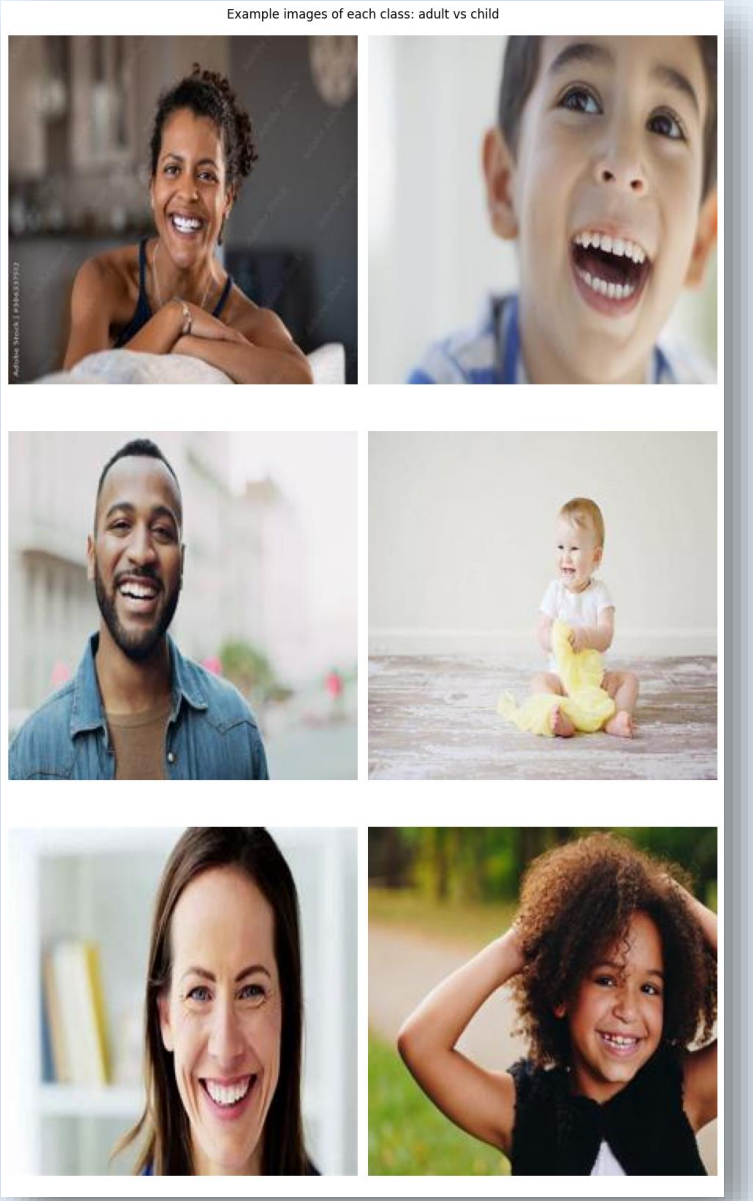
PoC: The dataset

Binary classification problem

2 classes labeled as: child and adult

522 images of each class:

- Total of images in test folder: 209
- Total of images in train folder: 668
- Total of images in validation folder: 167



PoC: The dataset

The sources of the dataset were images from the Imsitu dataset and Google Image Scraper

From the Imsitu dataset I was able collect 113 images of adults and 31 of children. I was able to do so with python functions and the "`imsitu_space.json`" and "`train.json`" files. I used the verb "smiling" and a series of codes referring to nouns such as "man", "woman" and "baby"

From the google image scraper I collected 409 images of adults using the search keywords "adult smiling" and 491 images of children with the search keywords "child smiling"

PoC: Preprocessing

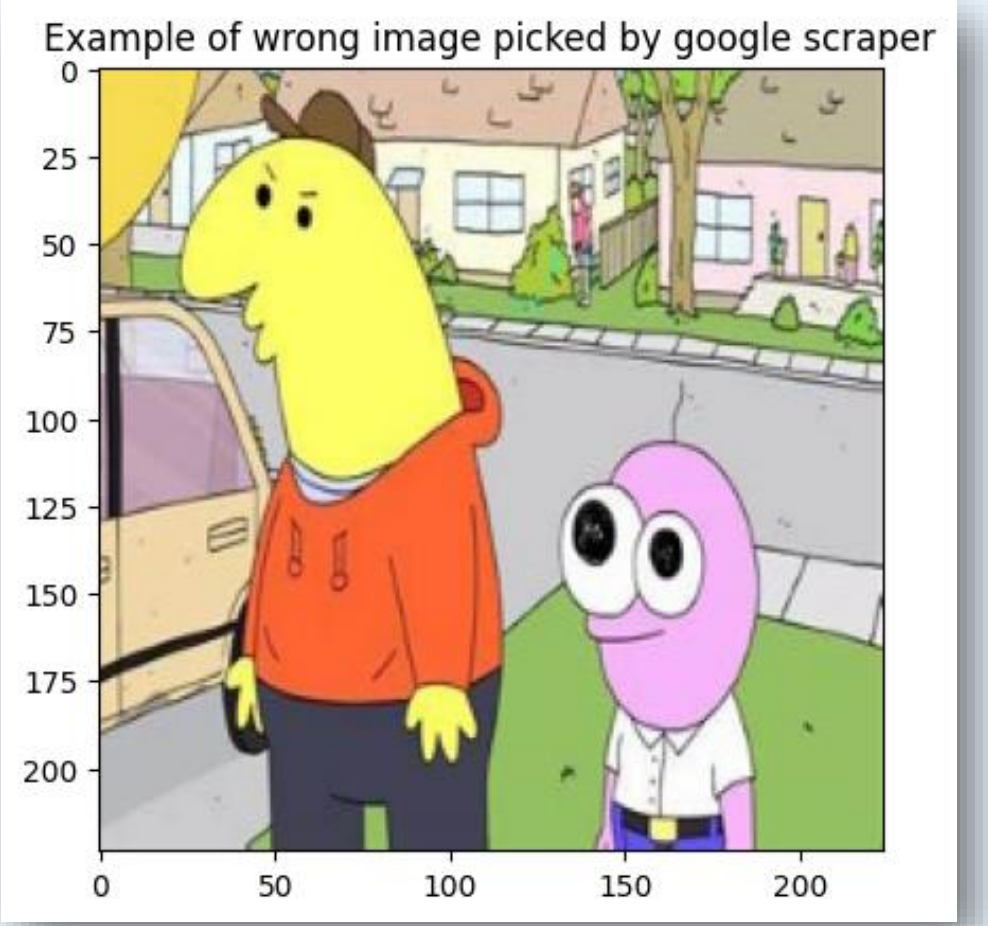
- Data cleaning :

I had to manually look at the scraped images to find and delete any images that didn't match the intended adult or child images.

- Data balancing:

This is the process of equalizing the number of images in each class to avoid bias towards any class.

I ended up with 522 images per class



PoC: Preprocessing

- Image resizing:

Resizing all the images in the dataset to the same size:

- 224x224

Image resizing.ipynb

```
# Set the target size for the resized images
target_size = (224, 224)

# Loop through all the files in the source directory
for filename in os.listdir(source_dir):
    # Open the image using Pillow
    img = Image.open(os.path.join(source_dir, filename))

    # Convert the image to RGB mode
    img = img.convert('RGB')

    # Resize the image to the target size
    img_resized = img.resize(target_size)

    # Save the resized image to the destination directory
    img_resized.save(os.path.join(dest_dir, filename))
```

Snipped

PoC: Preprocessing

- Data splitting:

Dividing the dataset into training, validation, and testing sets.

Data Splitting.ipynb

```
# Split your data into training and testing sets
train_df, test_df = train_test_split(df_complete, test_size=0.2, stratify=df_complete['age_group'], random_state=100)

# Split your training set into training and validation sets
train_df, val_df = train_test_split(train_df, test_size=0.2, random_state=100)

# Copy the image files into the appropriate folders
for index, row in train_df.iterrows():
    source = 'C:/Users/franc/OneDrive/Ambiente de Trabalho/Block C/User-case 3 - Correct dataset/data/resized_images/' + row['file_name']

    if row['age_group'] == 'child':
        dest = 'data/train/child/' + row['file_name']
    else:
        dest = 'data/train/adult/' + row['file_name']
    shutil.copyfile(source, dest)
```

Snipped

PoC: Preprocessing

- Data normalization and Data augmentation:

Scaling down the pixels so that they fall within a specific range in this case 0 and 1.

Additionally, data augmentation techniques such as rotation, zooming, and flipping were applied

This generates new training data and improve the model's ability to generalize

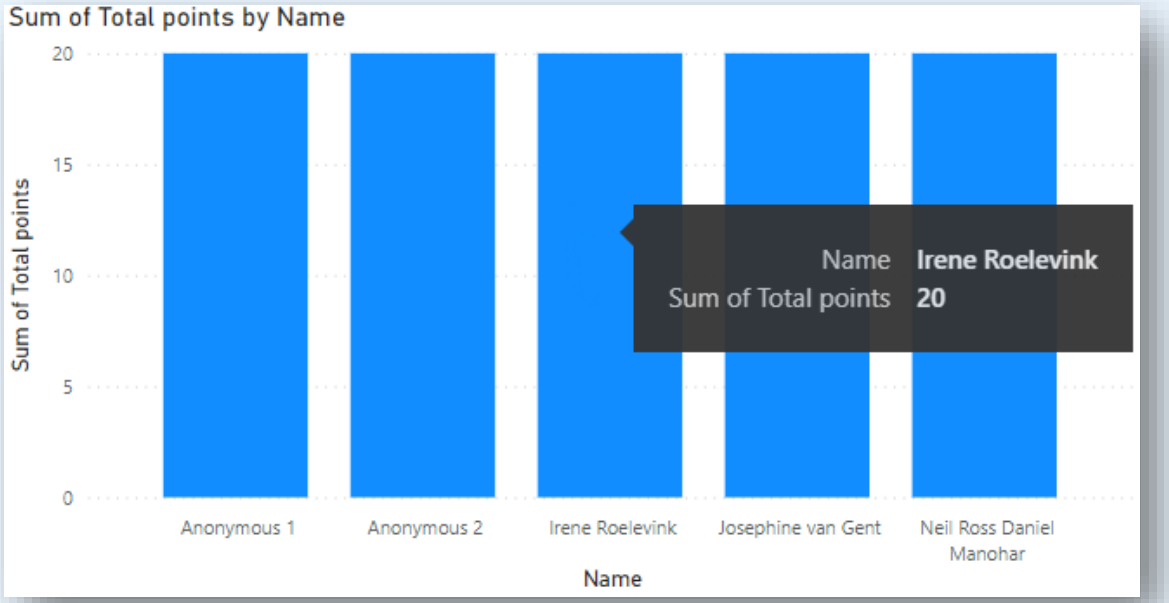
Data normalization.ipynb

```
# Create an instance of the ImageDataGenerator class
# with data augmentation parameters
train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=20,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)

# Load the images from the directory and apply data augmentation
train_generator = train_datagen.flow_from_directory(
        'data/train',
        classes=['child', 'adult'],
        target_size=(224, 224),
        batch_size=32,
        class_mode='binary')

# Load the validation and test sets without data augmentation
valid_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

val_generator = valid_datagen.flow_from_directory(
        'data/val',
        classes=['child', 'adult'],
        target_size=(224, 224),
        batch_size=32,
        class_mode='binary')

test_generator = test_datagen.flow_from_directory(
        'data/test',
        classes=['child', 'adult'],
        target_size=(224, 224),
        batch_size=32,
        class_mode='binary')
```

Snipped

PoC: Baselines

- Random guess accuracy?
  - Binary classification problem: 50%
- Human-Level Performance?
  - Based on the forms responses: 100%
- Multilayer Perceptron?
  - Test accuracy: 49,76%




```
# Evaluate the model on the test data
test_loss, test_acc = model_mlp.evaluate(test_generator)
```

```
Found 209 images belonging to 2 classes.
7/7 [==============================] - 1s 131ms/step - loss: 0.6932 - accuracy: 0.4976
```

PoC: Baselines

- Random guess accuracy?
  - Binary classification problem: 50%
- Human-Level Performance?
  - Based on forms responses: 100%
- Multilayer Perceptron?
  - Test accuracy: 49,76%

Average performance of baselines:

66,58

PoC: CNN architecture

- VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 conv2d_2 (Conv2D)           (None, 5, 5, 32)          147488

 max_pooling2d_2 (MaxPooling  (None, 2, 2, 32)         0
 2D)

 flatten_2 (Flatten)         (None, 128)               0

 dense_4 (Dense)             (None, 256)               33024

 dropout_2 (Dropout)         (None, 256)               0

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 14,895,457
Trainable params: 180,769
Non-trainable params: 14,714,688
_________________________________________________________________
```

PoC: CNN architecture

- I added a convolutional layer with 32 filters and a max pooling layer, followed by a flatten layer, two dense layers with 256 and 1 neurons respectively, and a dropout layer with a rate of 0.5.

- The VGG16 model is then frozen by setting its trainable attribute to False. This means that the weights of the VGG16 model will not be updated during training of the new model, only the weights of the new layers will be updated.

```
# Creating a VGG16 model
base_model = VGG16(weights='imagenet',
                   include_top=False,
                   input_shape=(224, 224, 3))

# Creating a new model on top of the VGG16 model
model_vgg16 = Sequential()
model_vgg16.add(base_model)
# add one convolutional and max pooling layer here
model_vgg16.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
model_vgg16.add(MaxPooling2D((2, 2)))
model_vgg16.add(Flatten())
model_vgg16.add(Dense(256, activation='relu'))
model_vgg16.add(Dropout(0.5))
model_vgg16.add(Dense(1, activation='sigmoid'))

# Freezing the VGG16 model
base_model.trainable = False
```

PoC: CNN training

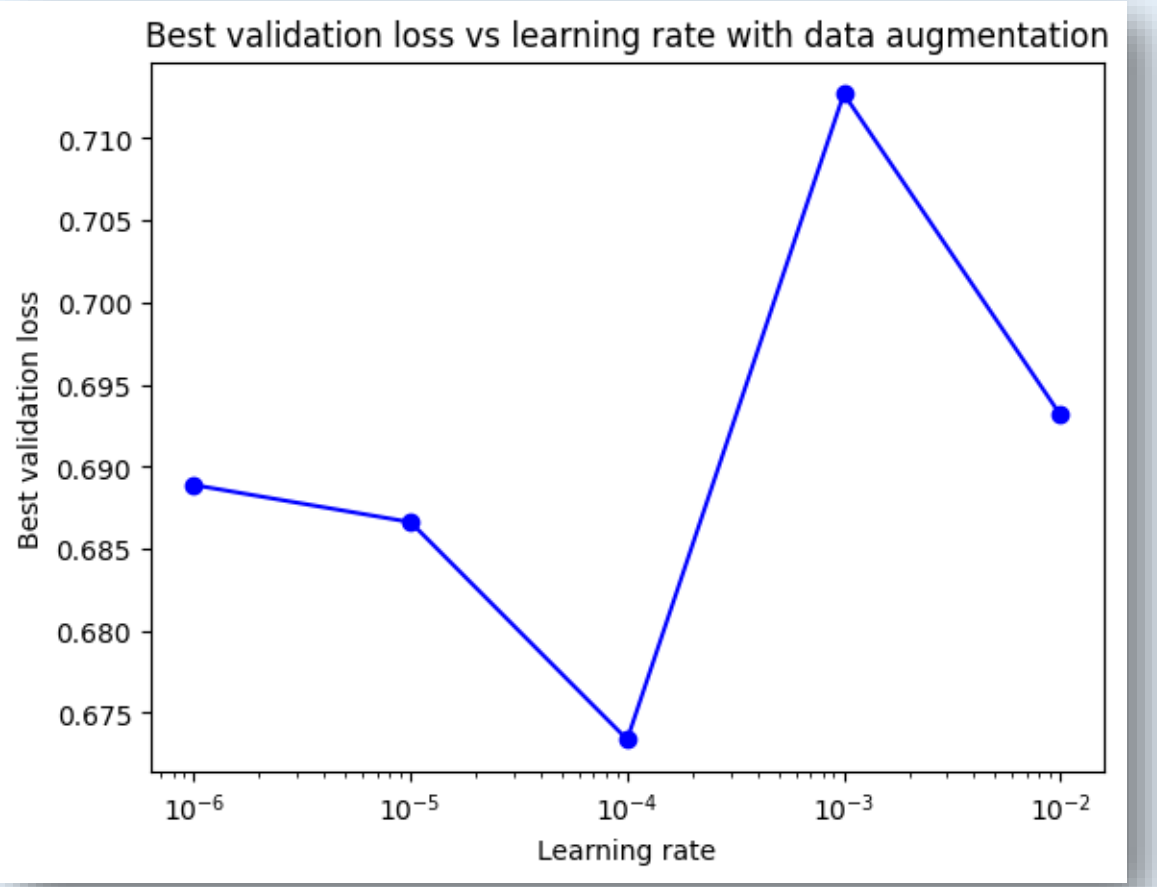
- I used binary crossentropy as loss function since the output of my model is binary and this function measures the difference between two probability distributions, which are the predicted output and the true output

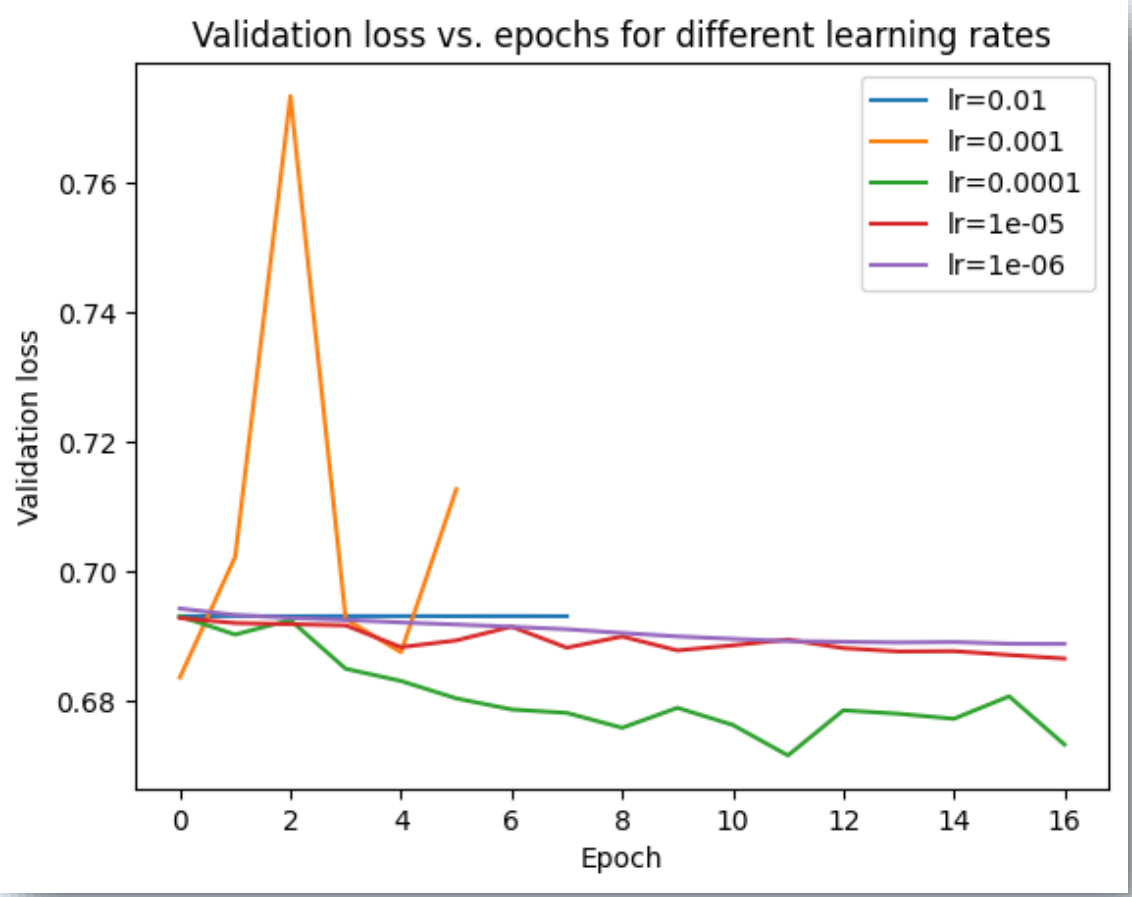
- I used Adam optimizer with a learning rate of 0.0001 since it performed better compared to other learning rates.

- For metrics I only used accuracy since it's a binary classification problem, the accuracy is enough to take the necessary conclusions of the model's performance

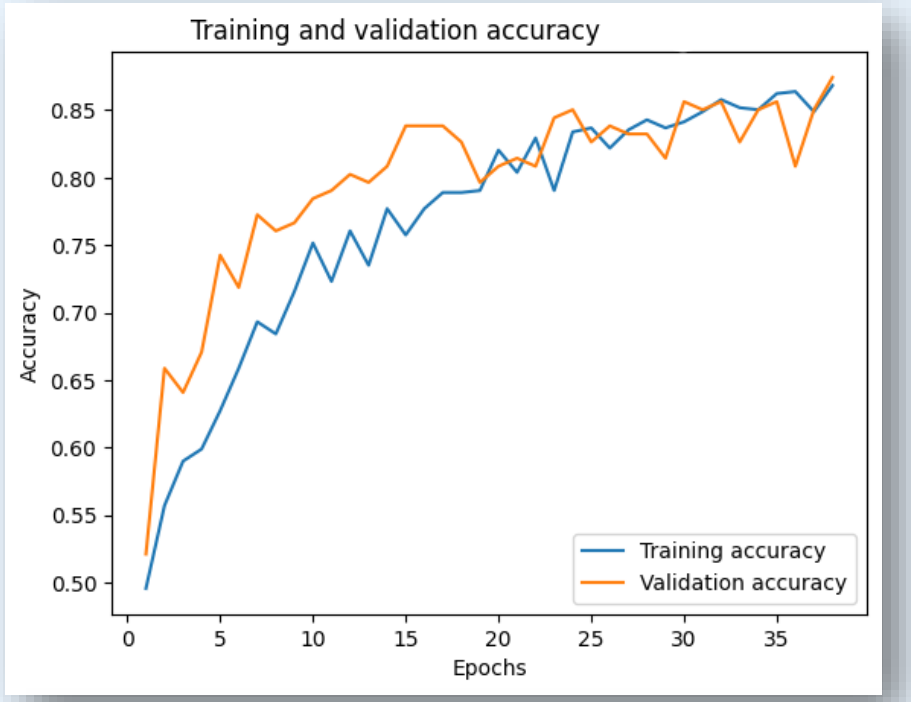
- I trained my model for 60 epochs but I added Early Stopping to monitor the validation loss, with a patience of 3

PoC: CNN training





PoC: Model performance

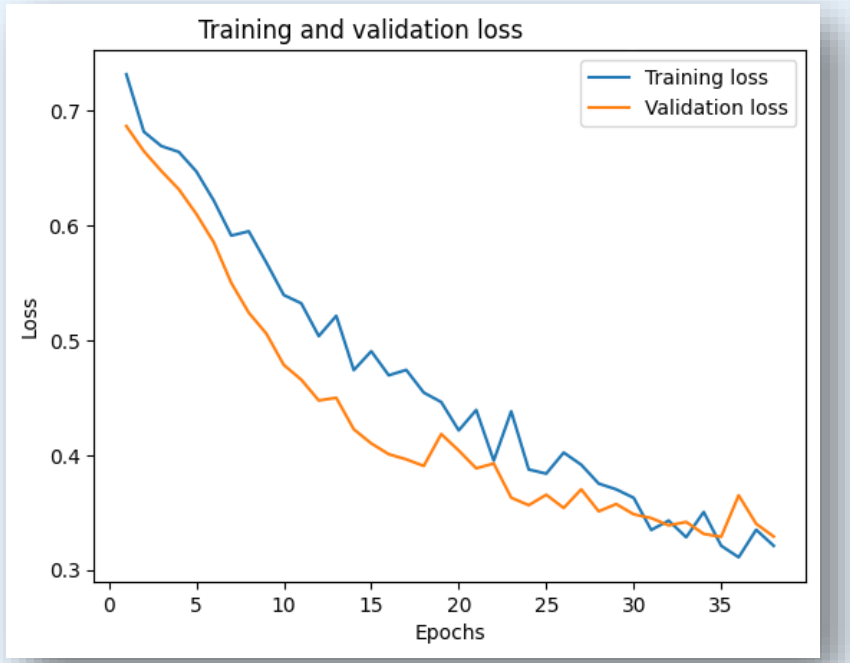



```
best_val_accuracy = max(H.history['val_accuracy'])
best_val_accuracy
```

```
0.8742514848709106
```

- Underfitting: Initial stages (epochs 1-10) low training and validation accuracy with gradual increase, suggesting possible underfitting due to low learning rate (0.0001) not model complexity (VGG16 pre-trained with additional layers).
- Overfitting: Around epochs 10-20, training accuracy increases while validation accuracy plateaus or decreases, indicating overfitting. Model may be too complex, fitting noise in training data, resulting in poor performance on validation set.
- Gap between Training and Validation Accuracy: Visible gap between training and validation accuracy throughout training process, with training accuracy consistently higher. Indicates potential overfitting, as model performs better on training data compared to unseen data.

PoC: Model performance

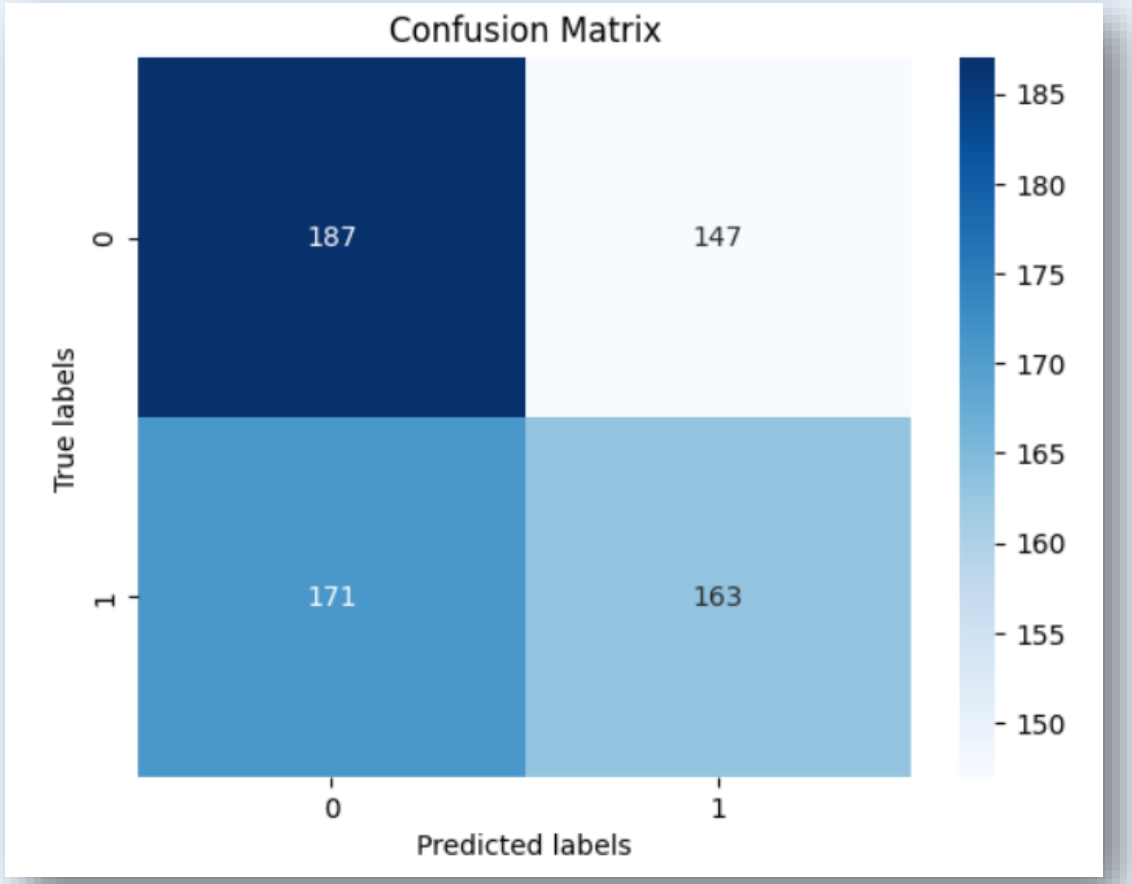



```
# Evaluating the model on the test set
test_loss, test_acc = model_vgg16.evaluate(test_generator, verbose=1)
print('Test accuracy:', test_acc)
```

```
7/7 [==============================] - 1s 82ms/step - loss: 0.4459 - accura
Test accuracy: 0.7942583560943604
```

- Epochs 1-10: Training and validation loss are initially high and close, indicating underfitting.
- Epochs 10-20: Losses decrease steadily, suggesting improving performance.
- Epochs 20 onwards: Training loss keeps decreasing, while validation loss plateaus or increases, indicating overfitting.
- Gap between training and validation loss widens, indicating increasing overfitting.
- Overfitting becomes more severe as training continues, leading to poor performance on the validation set.

PoC: Error analysis



PoC: Error analysis

True label: [1.], Predicted label: 0.0, Probability: 0.47

```
Misclassified probabilities:
[0.47357595]
```

PoC: Iteration

Models: first model → first iteration → second iteration/final model

In the first model I built the architecture from scratch, and it was quite similar to the layers I added to my final VGG16 model, the main differences being the number of units of the convolution layer and the number of units on my dense layer. The model's best validation accuracy was of 56% but the test accuracy ended up being 50,7%

- Because the accuracy barely surpassed the random guess accuracy, the loss was high and the model was overfitting I decided to apply a pre-trained model: VGG16

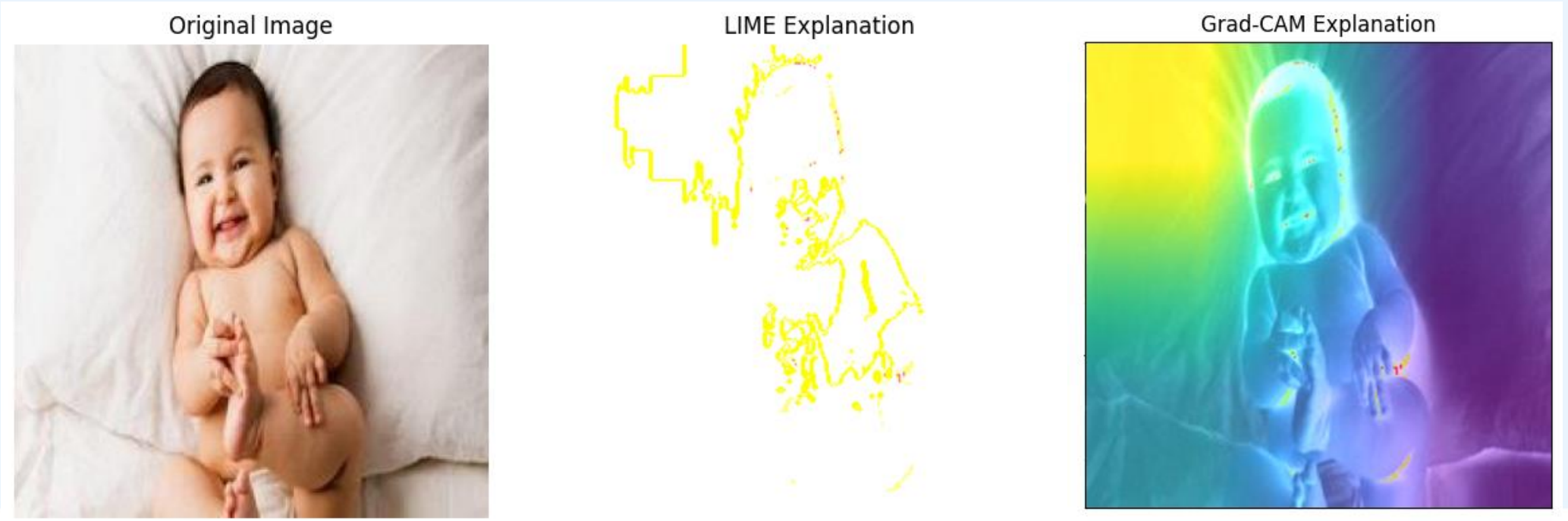
On my first attempt of using VGG16 I did not use data augmentation. The only difference from this one's architecture to the final model was a lack of a convolutional max pooling layer. The model's best validation accuracy was 83% and the test accuracy was 78%

- Despite the great improvement on accuracy and the big diminishment of the validation loss, when plotting the graphs to check the model performance I found out that the model was overfitting a lot since by the 10th epoch the training accuracy was already near 100% and the gap between training and validation loss was enormous too. On my final model, I applied data augmentation, added a convolutional max pooling layer and the results greatly improved as seen previously

Responsible AI (Transparency & Interpretability)

Bias, fairness, transparency and interpretability

My model's accuracy reached about 80% percent on a test set it had never seen before. It’s considerably high but we still have to know what’s driving my model’s decisions to make sure it’s not noise. Upon looking at LIME and Grad-CAM methods we can still that the model is picking on some important features such as the outline of the face and body but it is still picking considerably on some noise in the top left corner, indicating room for improvement.



Responsible AI (Transparency & Interpretability)

Bias, fairness, transparency and interpretability

Interpretability vs Accuracy trade-off

Overall, since we are talking about an algorithm that labels people into categories, interpretability should come first.

What good is a model which has high accuracy, but shows bias and misclassifies minority groups, or uses sensitive attributes such as gender and weight to make its decisions, it would be morally and ethically incorrect.

HCAI - Application Design:

High-level concept design:

- Simplicity is very important to make it as accessible as possible
- It’s supposed to be simple, other than a starting screen with languages and the algorithm which deploys the map it only has a privacy policy and a feedback section both of which are optional
- Clean color pallet consisting of mostly white and blue tones
- The app is to be deployed on a large tablet located near the entrance of the hospital, the patience is supposed to see it when they enter and quickly be informed of their destination with a few clicks

HCAI - Application Design

Algorithm Embedding:

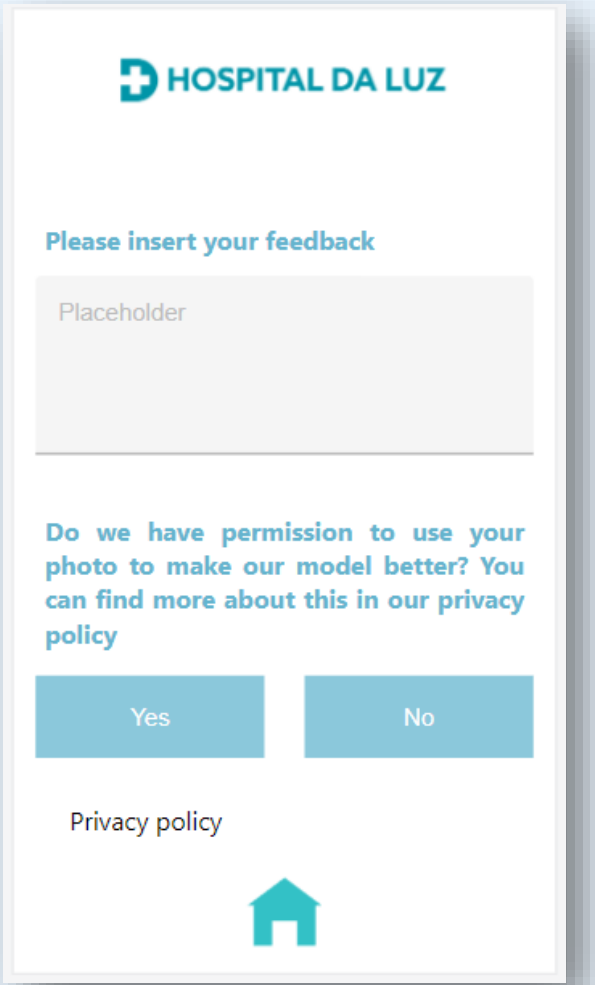
- Camera on the main screen that upon taking a photo of the patient
- It will triage them into a class (either adult or child) and give them a hospital map with directions accordingly
- At this point it does not do more than intended in the POC but there is the future possibility to make it so that it also recognizes patients and immediately directs them to their appointment

Breda University OF APPLIED SCIENCES

HCAI - Application Design:

Continuous Learning

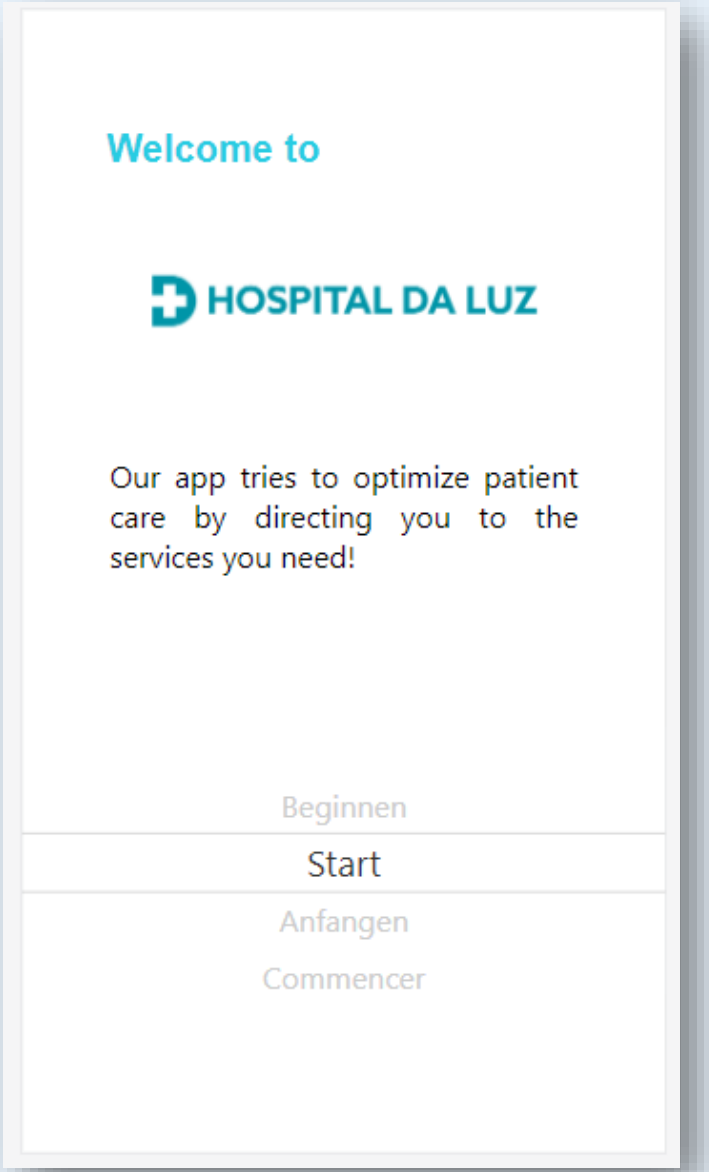
- Patients can contribute to help the model get better by giving permission for their information to be used



HCAI - Application Design:

Feedback

- User tests results?
  - It's a good idea to help reduce staff's workload
  - The feature to express when the model misclassified drastically improves the app's usability
- Client feedback?
  - Good value proposition
  - Nice and simple app design
- Design implications?
  - It might be redesigned with sound to be more accessible of blind people
  - Multiple language options to be inclusive of international patients



HCAI - Application Design:

Low-level design:

- Users can use the app with a few clicks
- They have the option to scan a QR code to keep the map on their phones
- They can choose to provide some feedback as well as authorize or not that their data is used to improve the model to learn continuously
- To improve accessibility language options are displayed on the start screen
- The prototype was designed to be as clear and easy to use as possible
- To be improved: implementing sounds to help make it more accessible to blind people

References

That were used for market research

- New York Times. “Hospitals Face Critical Shortage of Nurses and Other Workers.” August 21, 2021. https://www.nytimes.com/2021/08/21/health/covid-nursing-shortage-delta.html.

- American Hospital Association. “Data Brief: Health Care Workforce Challenges Threaten Hospitals’ Ability to Care.” November 1, 2021. https://www.aha.org/fact-sheets/2021-11-01-data-brief-health-care-workforce-challenges-threaten-hospitals-ability-care.

- Forbes. “The Healthcare Industry Is Crumbling Due To Staffing Shortages.” August 26, 2022. https://www.forbes.com/sites/saibala/2022/08/26/the-healthcare-industry-is-crumbling-due-to-staffing-shortages/?sh=74725dd07d6e.

- World Health Organization. “Task Shifting to Tackle Health Worker Shortages.” 2008. https://www.ncbi.nlm.nih.gov/books/NBK148518/.

- PyImageSearch. “OpenCV Age Detection with Deep Learning.” April 13, 2020. https://pyimagesearch.com/2020/04/13/opencv-age-detection-with-deep-learning/

- World Health Organization. “Nursing and Midwifery.” May 2020. https://www.who.int/news-room/fact-sheets/detail/nursing-and-midwifery.