Jorge Eliecer Benitez

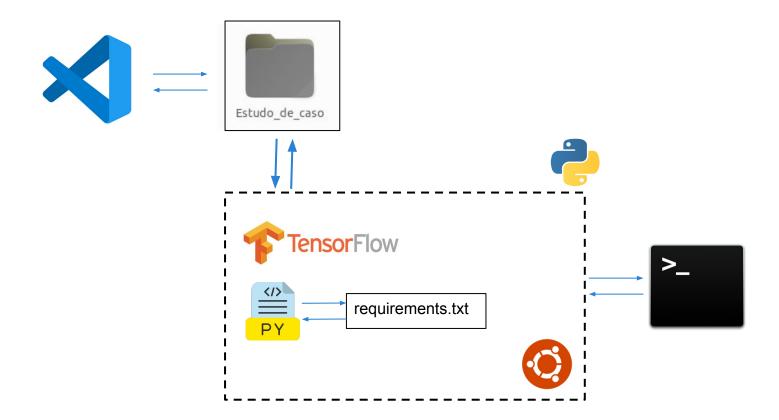
Prova Prática - 00577\_2024

Florianópolis, 03 de abril de 2024

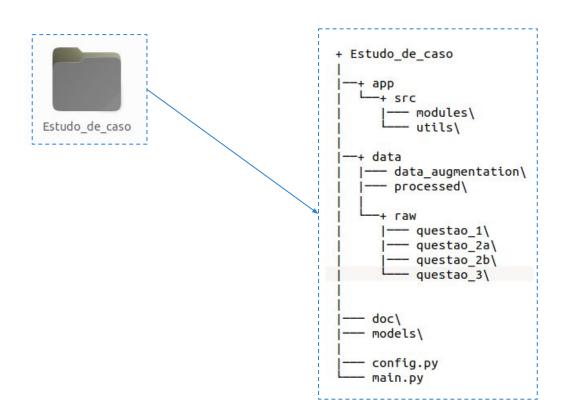
#### Tópicos.

- Segmentação por meio de métodos clássicos.
- Segmentação por meio de aprendizagem profunda.
- Gerenciamento de bandas espectrales.
- Uso de índices espectrais.
- Gerenciamento de rasters.
- Georreferenciamento.

#### Ambiente de desenvolvimento.



#### Estrutura das pastas.



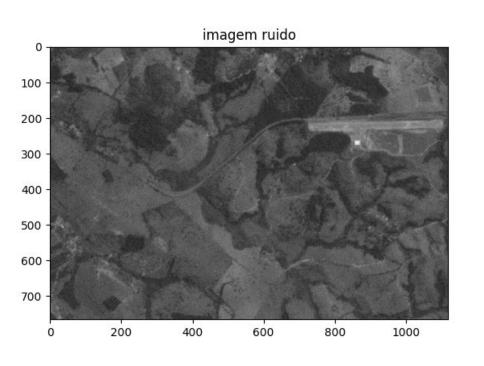
# Questão 1. - Contexto do problema.

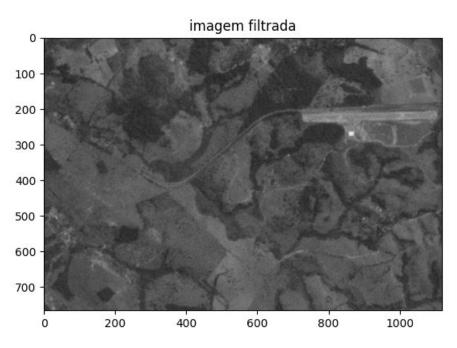
Código com alguma técnica para remoção parcial do ruído.



*salt & pepper (PSNR=25dB)* 

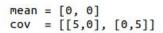
# Questão 1. - Resultados qualitativos / kernel\_1 [3 X3].

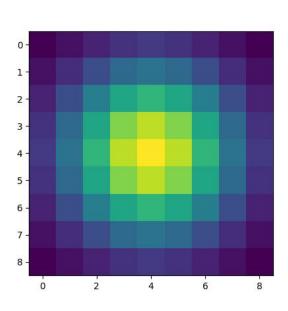


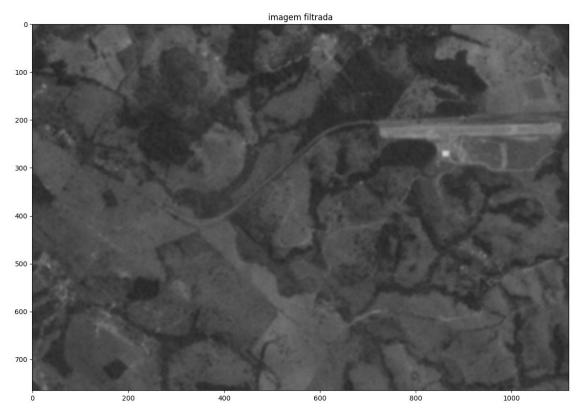


# Questão 1. - Resultados qualitativos / kernel\_gaussiano.

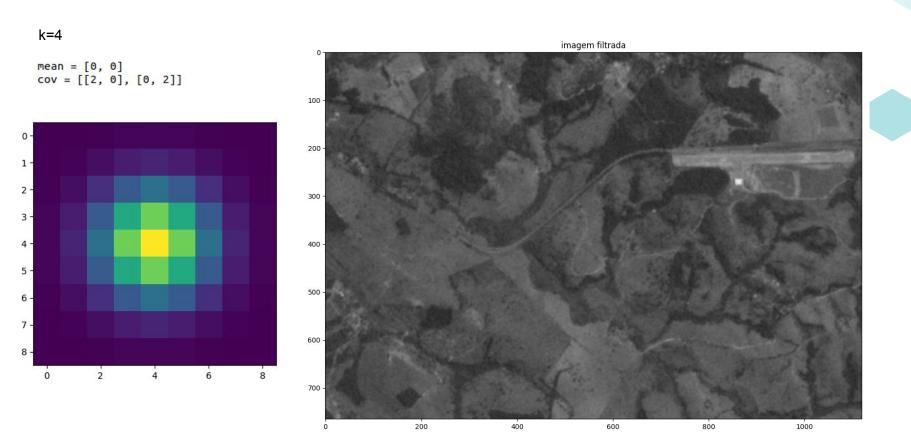
k=2





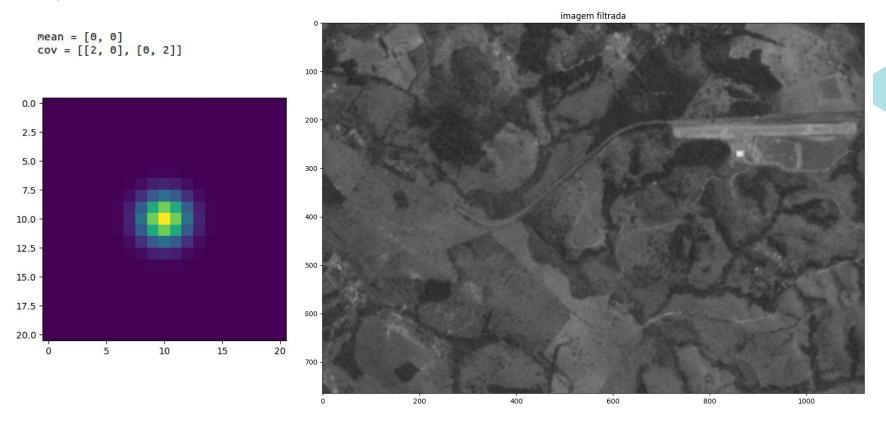


# Questão 1. - Resultados qualitativos / kernel\_gaussiano.



# Questão 1. - Resultados qualitativos / kernel\_gaussiano.



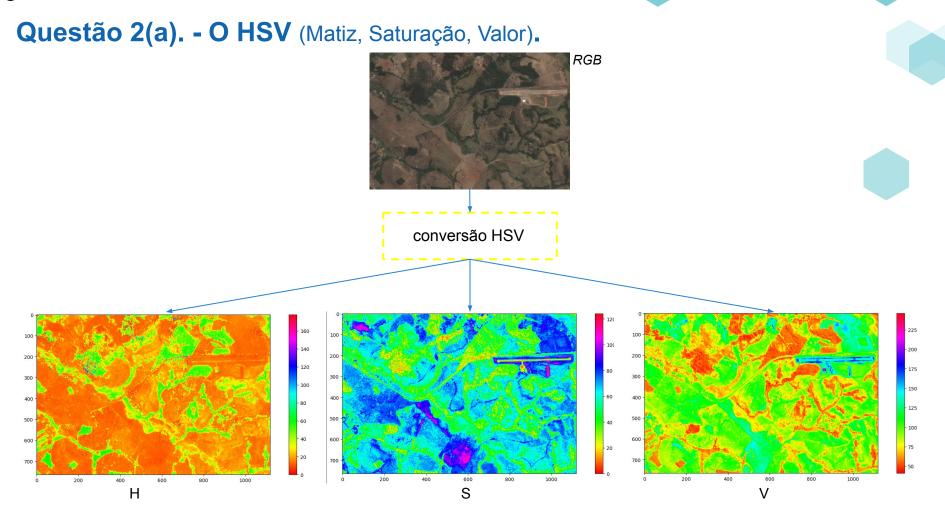


# Questão 2(a). - Contexto do problema.

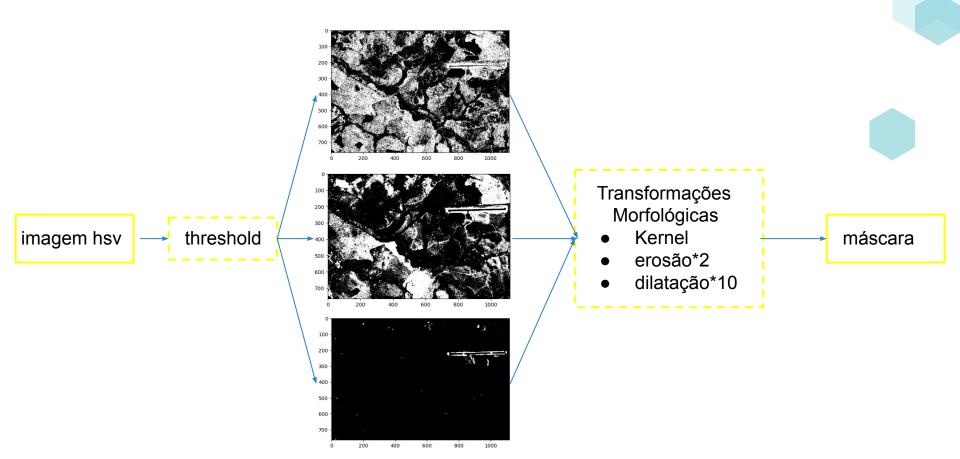
Considerando a imagem RGB airport.png apresenta uma pista de aeroporto. Implemente um código responsável pela seleção dos conjuntos de pixels associados ao objeto aeroporto.

.



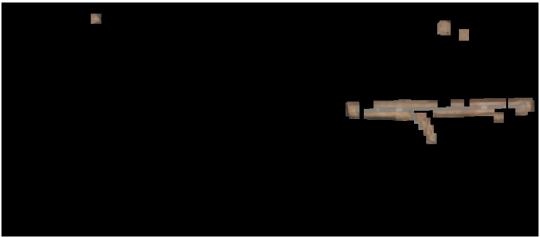


#### Questão 2(a). - Segmentação HSV / pipeline.



# Questão 2(a). - Resultados qualitativos.



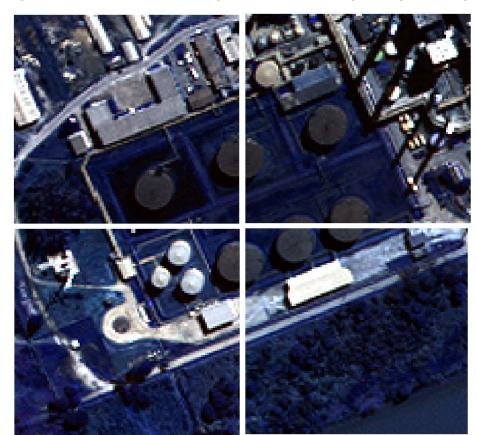


# Questão 2(b). - Contexto do problema.

Considerando um conjunto de imagens satelitais, implemente algum método para segmentação da vegetação.

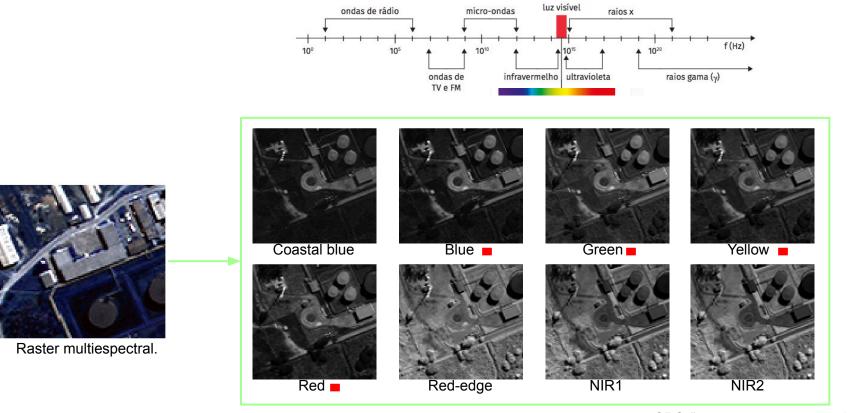
espectro do
raster WorldView 2. ref[1]

Description	Band ID
Coastal blue	1
Blue	2
Green	3
Yellow	4
Red	5
Red-edge	6
NIR1	7
NIR2	8



representação em falsa cor.

# Questão 2(b). - Índices espectrais.



CRS "componente geoespacial"

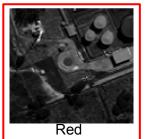
# Questão 2(b). - Índices espectrais.



Raster multiespectral.

#### Processamento espectral



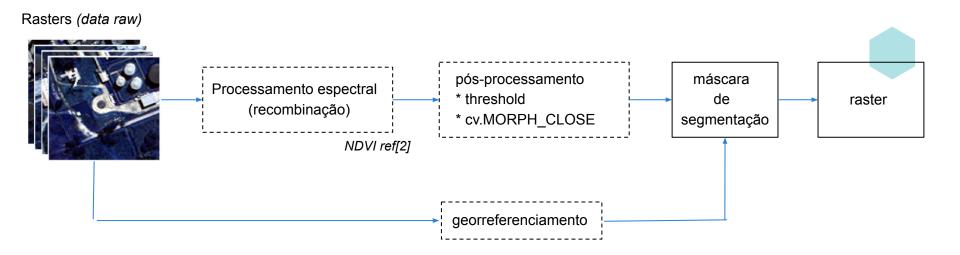




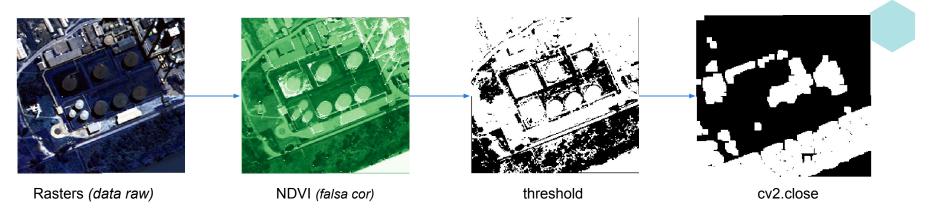


Raster RGB.

#### Questão 2(b). - Segmentação 2(b) / Pipeline.



# Questão 2(b). - Segmentação 2 / Pipeline.



# Questão 2(b). - Resultados qualitativos / kernel\_1.



Threshold = >0.25



Threshold = >0.5

```
[[1 1 1 1 1 1 1 1]

[1 1 1 1 1 1 1 1]

[1 1 1 1 1 1 1 1]

[1 1 1 1 1 1 1 1]

[1 1 1 1 1 1 1 1]

[1 1 1 1 1 1 1 1]

[1 1 1 1 1 1 1 1]
```

kernel

# Questão 2(b). - Resultados qualitativos / kernel\_2.



Threshold = >0.25



Threshold = >0.5

```
[[0 0 0 1 0 0 0]

[0 1 1 1 1 1 0]

[1 1 1 1 1 1 1]

[1 1 1 1 1 1 1]

[1 1 1 1 1 1 1]

[0 1 1 1 1 1 0]

[0 0 0 1 0 0 0]]
```

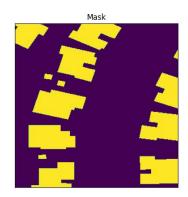
kernel

#### Questão 3. - Contexto do problema.

implemente algum modelo responsável pela segmentação das residências presentes as imagens satelitais disponibilizadas.









#### Questão 3. - Estrutura das pastas.



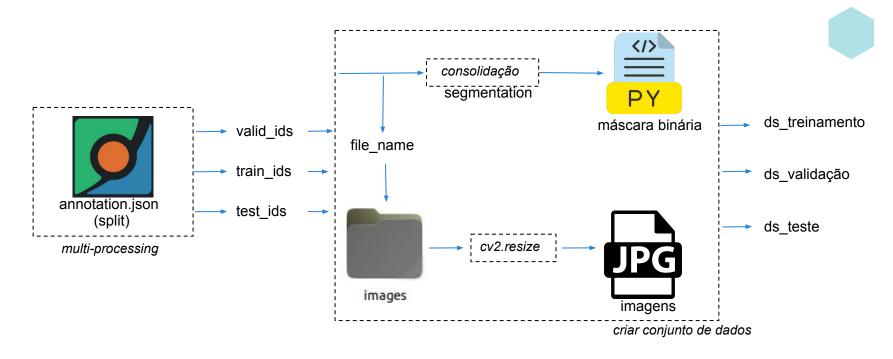
```
+ Estudo de caso
    -+ STC
          + modules
             -+ UNet model
               — dataloader.py
               - dataset.py
                  model.py
                - train.pv
         -utils\
 -+ data
      - data augmentation\
      - processed\
        гам
        -+ train
            - images\
           annotation. json
           annotation-small.json
 --- doc\
  -- models\
    config.py
    main.py
```

```
DATA_DIR = f'{os.getcwd()}/data/raw/'
MODEL_DIR = f'{os.getcwd()}/models/'
MODEL_LOAD = f'{MODEL_DIR}best_model/'

BACKBONE = 'resnet50'
BATCH_SIZE = 1
CLASSES = ['buildings']
EPOCHS = 40
LR = 0.0001
Re_FIT = 1
SPLIT = [70, 30]
```



#### Questão 3. - O conjunto de dados / pipeline.



# Questão 3. - O conjunto de dados / resumo.

```
+ UNet_model
|--- dataloader.py
|--- dataset.py
|--- model.py
|--- train.py
```

#### data augmentation

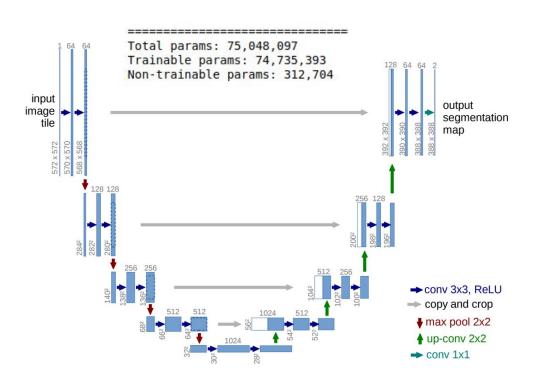
train\_transform = [
A.HorizontalFlip(p = 0.5),

Número de classes	Número de imagens	Tamanho da imagem	Número de anotações 71871	
1	8366	[300, 300, 3]		
1	280741	[300, 300, 3]	2395553	

A.HorizontalFlip( $p = 0.5$ ),
A. $VerticalFlip(p = 0.5)$ ,
A.Rotate(limit=[-30,30],)

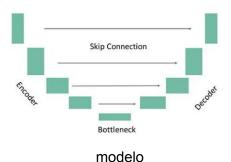
#### Questão 3. - O modelo.

```
+ UNet_model
|--- dataloader.py
|--- dataset.py
|--- model.py
|--- train.py
```



#### Questão 3. - O treinamento / hiperparâmetros.

```
+ UNet_model
|--- dataloader.py
|--- dataset.py
|--- model.py
|--- train.py
```



#### segmentation\_models

```
BACKBONE = 'resnet50'
BATCH SIZE = 1
CLASSES = ['prédios']
EPOCHS = 40
LR = 0.0001
Re FIT = 1
SPLIT = [70, 30]
BACKBONE = 'efficientnetb7'
BATCH SIZE = 1
CLASSES = ['prédios']
EPOCHS = 40
LR = 0.0001
Re FIT = 1
SPLIT = [70, 30]
annotation-small.json
BACKBONE = 'resnet50'
BATCH SIZE = 1
CLASSES = ['prédios']
EPOCHS = 3
LR = 0.0001
Re FIT = 1
```

Tempo X época: ~14m : 40s

Tempo X época: 6h : 42m :30s

\*Backbones ref[2]

SPLIT = [70, 30]

annotation.json

#### Questão 3. - O treinamento / recursos usados [320, 320, 3].

Unet-resnet50

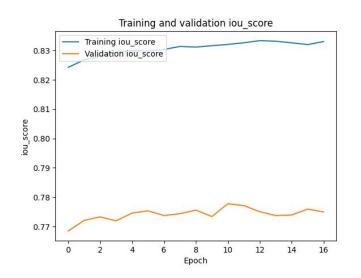
NVID	IA-SMI	550.54.14	4		Driver V	ersion:	550.54	4.14	CUDA Versio	n: 12.4
GPU Fan	Name Temp	Perf				Bus-Id				Uncorr. ECC   Compute M.   MIG M.
0 N/A	NVIDIA 80C	GeForce P0		59W /	Off   80W   	50901	MiB /		   51%   +	N/A
Proce GPU	esses: GI ID	CI ID	PID	Туре	Process	name				GPU Memory   Usage
0 0	N/A N/A		2125 399095	G C	/usr/li python	.b/xorg/	(org	 		=======  4MiB   5082MiB

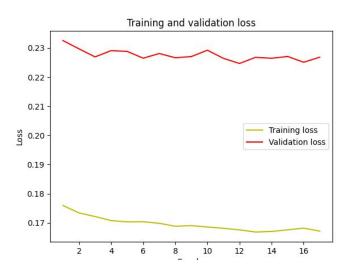
```
0[|| 6.0%] 3[|||||| 20.9%]
1[|||| 11.1%] 4[|| 3.9%]
2[|||||| 19.2%] 5[|||| 9.2%]
Mem[|||||| 19.2%] 5[|||| 9.906/15.46]
Swp[||||||| 17.1%] 9[|| 4.9%]
7[|||| 9.3%] 10[|||| 10.4%]
8[||| 6.5%] 11[|||||| 24.2%]
Tasks: 182, 1481 thr; 3 running
Load average: 1.84 1.54 1.27
Uptime: 1 day, 01:52:13
```

Unet-efficientnetb7

NVIDI	IA-SMI	550.54.1	4		Driver	Version: 550.	54.14	CUDA Versio	n: 12.4
GPU Fan	Name Temp	Perf		Persiste Pwr:Usag		Bus-Id Mer			Uncorr. ECC Compute M. MIG M.
0 N/A	NVIDIA 71C	GeForce P0		====== 60 Ti 77W /	Off 80W		1:00.0 Off 6144MiB	   95% 	N/A Default N/A
Processes: GPU GI CI PID Type Proce ID ID					Proces	ss name			GPU Memory Usage
0 0	N/A N/A		====== 2125 394621	 G C	usr/l pythor	========= Lib/xorg/Xorg 1	=======	=======	 4MiB 5086MiB

#### Questão 3. - Resultados quantitativos resnet50 [608, 608, 3].



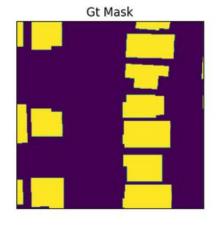


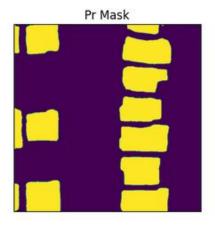
Loss: 0.22664

mean iou\_score: 0.77673 mean f1-score: 0.85879

# Questão 3. - Resultados qualitativos.

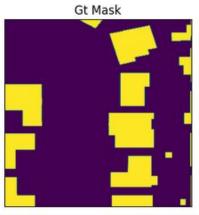


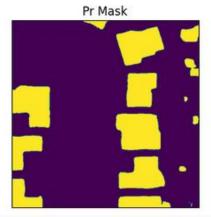














id:978

#### Referencias.

- Ref[1] WorldWiev2, https://earth.esa.int/eogateway/missions/worldview-2
- Ref[2] Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches, Omar Elharrouss and Younes Akbari and Noor Almaadeed and Somaya Al-Maadeed, 2022, arXiv.

#### Anexos / Anexo 1.

Ref[1] Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches.



**Figure 9:** Used backbones for each task. In some tasks some backbones are used more than the others and are illustrated with large scale in the figure.