UNIVERSIDADE FEDERAL DO AMAPÁ

INTELIGÊNCIA ARTIFICIAL

LUCAS VINÍCIUS KOGA SOUZA

RELATÓRIO SOBRE A IDENTIFICAÇÃO DE PEIXES AMAZÔNICOS

MACAPÁ

2021

Identificação de Peixes Amazônicos

1. Primeiro foi configurado o Bing Search API para realizar a busca das imagens, em seguida foi usado os comandos para armazenar nos diretórios:

```
fish_types = 'pirarucu','filhote','tucunaré','tambaqui','piranha'
path = Path('fish')
if not path.exists():
    path.mkdir()
    for o in fish_types:
        dest = (path/o)
        dest.mkdir(exist_ok=True)
        results = search_images_bing(key, f'{o} fish')
        download_images(dest, urls=results.attrgot('contentUrl'))
```

2. Em seguida, para remover possíveis imagens corrompidas que foram baixadas usamos o comando:

```
failed = verify_images(fns)
failed
```

```
failed.map(Path.unlink);
```

Depois disso, foi criado um DataLoaders:

```
fish = DataBlock(
    blocks=(ImageBlock, CategoryBlock),
    get_items=get_image_files,
    splitter=RandomSplitter(valid_pct=0.2, seed=42),
    get_y=parent_label,
    item_tfms=Resize(128))
```

3. Com isso definimos a resolução das imagens para 128px, o modelo Restnet18 e o fine_tune = 5 para realizar nosso treinamento

```
fish = fish.new(
    item_tfms=RandomResizedCrop(128, min_scale=0.5),
    batch_tfms=aug_transforms())
dls = fish.dataloaders(path)
```

```
learn = cnn_learner(dls, resnet18, metrics=error_rate)
learn.fine_tune(5)
```

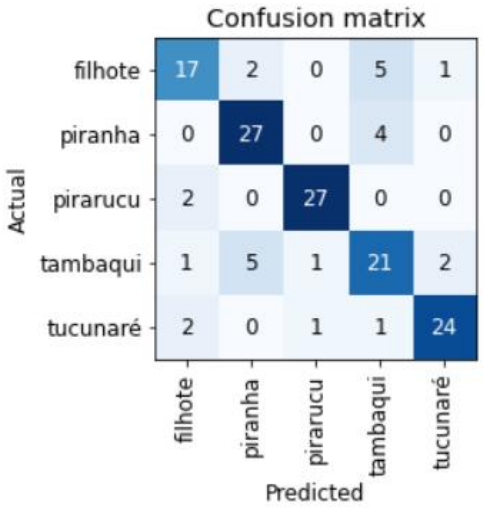
E obtivemos essa tabela:

epoch	train_loss	valid_loss	error_rate	time
0	2.448771	1.927915	0.440559	00:05

epoch	train_loss	valid_loss	error_rate	time
0	1.511709	1.103854	0.335664	00:05
1	1.296027	0.859734	0.251748	00:05
2	1.089121	0.639344	0.230769	00:05
3	0.916810	0.585916	0.209790	00:05
4	0.821855	0.558028	0.188811	00:06

4. Desse modo, conferimos como está o aprendizado através de uma matriz confusão, verificando em sua diagonal os casos acertados e ao redor os erros:

```
interp = ClassificationInterpretation.from_learner(learn)
interp.plot_confusion_matrix()
```



5. Para reduzir esses erros, executamos uma interface para fazer o tratamento da predição, ela nos permite ajustar ou excluir os erros encontrados:

```
cleaner = ImageClassifierCleaner(learn)
cleaner

for idx in cleaner.delete(): cleaner.fns[idx].unlink()
```

6. Com os erros minimizados e reduzidos, iniciamos a tentativa de tornar o notebook acessível em um site, para isso usamos os repositórios do GitHub e o Binder.