



UNIVERSIDADE FEDERAL DO AMAPÁ
ENGENHARIA ELÉTRICA
LUCAS VINÍCIUS KOGA SOUZA

RELATÓRIO SOBRE A IDENTIFICAÇÃO DE PEIXES AMAZÔNICOS

MACAPÁ

2021

LUCAS VINÍCIUS KOGA SOUZA

RELATÓRIO SOBRE A IDENTIFICAÇÃO DE PEIXES AMAZÔNICOS

Trabalho apresentado à Coordenação do Curso de Ciências da Computação, da Universidade Federal do Amapá como requisito de avaliação da disciplina Inteligência Artificial, ministrada pelo professor Clay Palmeira.

MACAPÁ

2019

Identificação de Peixes Amazônicos

1. Introdução

Neste trabalho será especificado as etapas de execução de uma Machine Learning FastAi no modelo Resnet18, o modelo foi treinado para identificar 5 espécies de peixes amazônicos: Piraíba (filhote), Pirarucu, Tucunaré, Tambaqui e Piranha.

Desse modo, executamos esse aprendizado utilizando a plataforma Gradient Paperspace trabalhada pelo professor Clay na aplicação de suas aulas. Nessa plataforma criamos notebooks do Jupyter que utilizam de GPUs remotas para melhor desenvoltura do processamento de dados.

2. Treinamento do modelo

O nosso modelo foi baseado no capítulo 2 do template do Paperspace, utilizando toda a biblioteca FastAi que a plataforma disponibiliza. Depois configuramos o Bing Search API para realizar a busca das imagens, em seguida foi usado os comandos para armazenar nos diretórios:

```
fish_types = 'pirarucu','piraíba','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'))
```

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

```
failed = verify_images(fns)
failed

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

Depois disso, para tornar nossos dados aptos para serem usados no treinamento 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))

```

Com isso definimos a resolução das imagens para 128px, o modelo Resnet18 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	2.265101	1.408244	0.430556	00:05

epoch	train_loss	valid_loss	error_rate	time
0	1.152012	0.967369	0.347222	00:05
1	0.955728	0.909969	0.243056	00:06
2	0.800412	0.846379	0.215278	00:05
3	0.682970	0.789115	0.201389	00:05
4	0.586376	0.776720	0.194444	00:05

Essa é a Tabela da base de dados do nosso Learner. 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()

```

Confusion matrix

Actual	piranha	26	0	0	0	1
	pirarucu	2	20	6	1	2
	piraíba	1	0	28	2	0
	tambaqui	4	0	1	20	1
	tucunaré	3	1	2	1	22
	Predicted	piranha	pirarucu	piraíba	tambaqui	tucunaré

Como podemos analisar, a linha diagonal destacada em tonalidades de azul mais forte indicam os acertos do Learn e os demais espaços mostram os erros em que o aprendizado identificou como um tipo de peixe quando era outro.

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()
```

Com os erros minimizados e reduzidos, executamos o fine tune novamente para treinar o modelo e obtivemos a seguinte matriz confusão:

Confusion matrix

Actual	piranha	25	0	2	1	1
	pirarucu	0	21	4	1	0
	piraíba	1	2	26	3	1
	tambaqui	3	0	1	18	2
	tucunaré	2	3	1	0	24
	Predicted	piranha	pirarucu	piraíba	tambaqui	tucunaré

Esta ainda apresenta alguns erros, no entanto, foram minimizados a fim de obter maior probabilidade de acertos e melhor precisão no diagnóstico das imagens.

3. Conclusão

Esse trabalho nos permitiu ver a importância do tratamento e do pré-processamento de dados na identificação das imagens, assim como no desenvolvimento de um bom aprendizado, focando na correção de erros e minimização destes. Desse modo, conseguimos aumentar significativamente a porcentagem de acertos na execução do projeto e ter uma análise mais precisa das espécies de peixes amazônicos.