



Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática

Ano Letivo de 2019/2020

Classificação do Coberto e Uso do Solo Português com Imagens de Satélite e Técnicas de Aprendizagem Automática

LEI

André Ramalho A76387

Mariana Lino Costa - A78824

Vítor Gomes – a75362

Julho, 2020

Resumo

Este relatório contém todo o processo e explicação para a realização de um programa que classifica o coberto e uso de solo português. Desde a escolha do *dataset* e do classificador base, até à aplicação de várias técnicas de aprendizagem automática e a análise dos seus resultados.

Fez-se download do *dataset* no *Sentinel Hub*, o classificador base escolhido foi o COS e foram usadas várias bibliotecas do *eo-learn* para se relacionar todos os dados.

As técnicas de aprendizagem automática utilizadas foram um classificador da LGBM, uma rede neuronal treinada por pixel, uma rede neuronal convolucional de transferência – ResNet - com o acréscimo manual de algumas camadas, e uma rede neuronal convolucional totalmente feita de raiz. O classificador LGBM teve uma *Accuracy* de 72.5%, a rede neuronal por pixel, 65.1%, a CNN ResNet50 teve 66.2%, a ResNet34 70.3%, e por fim a CNN de raiz com o melhor resultado, 80.1% de *Accuracy*. Estes valores estão analisados com mais detalhe abaixo.

Após se comparar as imagens de previsão resultantes do treino de cada técnica de aprendizagem referida, notou-se que a que obteve melhores resultados foi a CNN feita raiz.

Índice

1.	Introdução	5
1.1.	Motivação e contextualização.....	6
2.	Base de dados e Classificador do Coberto e Uso do Solo.....	7
2.1.	Dataset	7
2.2.	Preparação do <i>Dataset</i>	9
2.3.	Classificador COS.....	10
3.	Técnicas de Aprendizagem Automática	13
3.1.	Classificador LGBM	13
3.2.	Rede Neuronal	13
3.3.	CNN	15
3.3.1.	ResNet50.....	15
3.3.2.	Resnet34	17
3.3.3.	Rede convolucional de raiz	17
4.	Validação.....	19
4.1.	Classificador LGBM	20
4.2.	Rede Neuronal – por pixel	21
4.3.	CNN	22
4.3.1.	ResNet50.....	22
4.3.2.	ResNet34.....	23
4.3.3.	Rede convolucional de raiz	24
5.	Análise de resultados.....	25
5.1.	Classificador LGBM	25
5.2.	Rede Neuronal – por pixel	27
5.3.	CNN	28
5.3.1.	ResNet50.....	28
5.3.2.	ResNet34.....	29
5.3.3.	Rede convolucional de raiz	30
5.4.	Comparação das previsões.....	31
6.	Conclusão e trabalhos futuros.....	32
7.	Referencias	33

Índice de Figuras

Figura 1 - Esquema do processo de classificação do coberto e uso do solo.	6
Figura 2 - Região Minho de Portugal dividida em bboxes de 5x5 km.....	7
Figura 3 - Demonstração de cada classe no COS 2015.	10
Figura 4 - Tabela que demonstra o número de amostras de cada classificação em cada versão do COS para cada ano.....	11
Figura 5 - Resultado da classificação pelo COS no Minho de Portugal.	12
Figura 6 - Gráficos da evolução da accuracy e loss ao longo das épocas para a rede neuronal.	14
Figura 7 - Gráficos da evolução da accuracy e loss da ResNet50 ao longo de 100 épocas.....	16
Figura 8 - Gráficos da evolução da accuracy e loss da ResNet34 ao longo de 100 épocas.	17
Figura 9 - Gráficos da evolução da accuracy e loss ao longo das épocas para a rede convolucional de raiz.	18
Figura 10 - Matriz confusão e a transposta do classificador LGBM.	20
Figura 11 - Quantidade de amostras para cada classe que foram usadas no treino no Classificador LGBM.....	20
Figura 12 - Matriz confusão e a transposta da rede neuronal por pixel.....	21
Figura 13 - Quantidade de amostras para cada classe que foram usadas no treino na Rede Neuronal por pixel.	21
Figura 14 - Matriz confusão e transposta da ResNet50.	22
Figura 15 - Quantidade de amostras para cada classe que foram usadas no treino na ResNet50.	22
Figura 16 - Matriz confusão e transposta da ResNet34.	23
Figura 17 - Quantidade de amostras para cada classe que foram usadas no treino na ResNet34.	23
Figura 18 - Matriz confusão e transposta da CNN de raiz.....	24
Figura 19 - Quantidade de amostras para cada classe que foram usadas no treino na CNNraiz.....	24
Figura 20 - Previsão da cobertura do solo pelo classificador LGBM.	25
Figura 21 - Comparação da classificação do solo real com a classificação prevista pelo classificador LGBM.....	26
Figura 22 - Previsão da cobertura do solo pela rede neuronal por pixel.	27
Figura 23 - Comparação da classificação do solo real com a classificação prevista pela rede neuronal por pixel.	27
Figura 24 - Previsão da cobertura do solo pela rede ResNet50.....	28
Figura 25 - Comparação da classificação do solo real com a classificação prevista pela rede ResNet50.	28
Figura 26 - Previsão da cobertura do solo pela rede ResNet34.....	29
Figura 27 - Comparação da classificação do solo real com a classificação prevista pela rede ResNet34.	29
Figura 28 - Previsão da cobertura do solo pela rede convolucional de raiz.....	30
Figura 29 - Comparação da classificação do solo real com a classificação prevista pela rede convolucional de raiz.....	30

1. Introdução

No âmbito da Unidade Curricular de Laboratórios de Informática do primeiro ano de mestrado do curso de Engenharia Informática da Universidade do Minho, surge o projeto de criação de um classificador do coberto e uso do solo Português com imagens de satélite, através de várias técnicas de Aprendizagem Automática (*Machine Learning*).

Este primeiro capítulo apresenta uma visão geral sobre este relatório e uma breve descrição de todo o processo até ao resultado final, para além disso, apresenta também uma contextualização ao problema e os objetivos da sua realização. Todo o processo está dividido em 4 fases.

A primeira fase está representada no segundo capítulo, que contém todo o processo acerca de como se obteve a base de dados e a base de classificação. Começou-se então por procurar e organizar um *dataset* que tivesse as imagens de satélite de Portugal. Usou-se o *Sentinelhub*, um mecanismo para processamento de *petabytes* de dados de satélite, esta plataforma está a ajudar desenvolvedores de aplicativos e programas de processamento de imagens como este. Torna o *Sentinel*, o *Landsat* e outras imagens de observação da Terra facilmente acessíveis para navegação, visualização e análise.

Devido a falta de recursos computacionais, houve a necessidade de reduzir o *dataset*, ou seja, se fazer este projeto apenas para a região do Minho de Portugal.

Numa fase seguinte houve a necessidade de arranjar uma forma de relacionar as imagens já recolhidas com a sua classificação de solo, usou-se a Carta de Uso e Ocupação (COS), uma cartografia temática que pretende caracterizar com detalhe a ocupação do solo do território de Portugal continental. Esta informação encontra-se em formato vetorial e possuiu uma unidade mínima cartográfica de 1 hectare e uma distância mínima entre linhas de 20 metros.

Após se obter uma classificação quase certa das imagens de satélite de Portugal, passa-se à segunda fase deste projeto (capítulo 3), onde se procurou usar vários métodos de aprendizagem automática para classificar e com o intuito de se arranjar o melhor classificador de coberto e uso de solo possível. Entre eles, um classificador LGBM, redes neuronais, incluindo redes convolucionais, de forma a treinar imagens, para futuramente classificar novas imagens de satélite.

A terceira fase é a fase de validação (capítulo 4) que é bastante importante para fases futuras de análise de resultados, precisão e eficácia da máquina. A validação permite que haja parâmetros de comparação entre métodos de aprendizagem automática.

O capítulo 5 é representante da fase final de todo o processo, a etapa de analisar os resultados obtidos pelas técnicas de aprendizagem automática. É através dessa análise que se pode concluir se é necessário alterar e melhorar o programa. Existem várias formas de o melhorar, e ao longo da elaboração do projeto, aplicou-se várias técnicas que não vão ser todas apresentadas neste relatório com o intuito de não ficar extenso. Exemplos de como se podem melhorar os resultados é através de experimentação de diferentes redes de aprendizagem automática ou até mesmo melhorando o *dataset*, fazendo *data augmentation* ou balanceando os dados. No caso, optou-se pela técnica de balanceamento de dados.

1.1. Motivação e contextualização

O trabalho centraliza toda a sua estratégia na produção e análise do uso e ocupação do solo somente num momento, no ano 2018. A construção de um sistema deste tipo de informação dispõe um conjunto de informações que cria novas oportunidades para o desenvolvimento de múltiplas investigações acerca deste tema em várias perspetivas.

É bastante interessante se perceber as mudanças no meio ambiente, como urbanização, desmatamento e expansão agrícola, ocorrem na escala das paisagens e impactam diretamente os processos do ecossistema.

Apesar de se ter decidido apenas classificar o solo do Minho de Portugal no ano 2018, o programa é facilmente adaptável para se fazer de outros países com mais dados e até se fazer em vários períodos de tempo de forma a se ver a evolução do solo, isto desde que se tenha recursos computacionais suficientes.

O objetivo deste projeto teve o seu foco em se arranjar bons resultados de classificação e não propriamente em se observar e analisar a evolução do coberto e uso do solo.

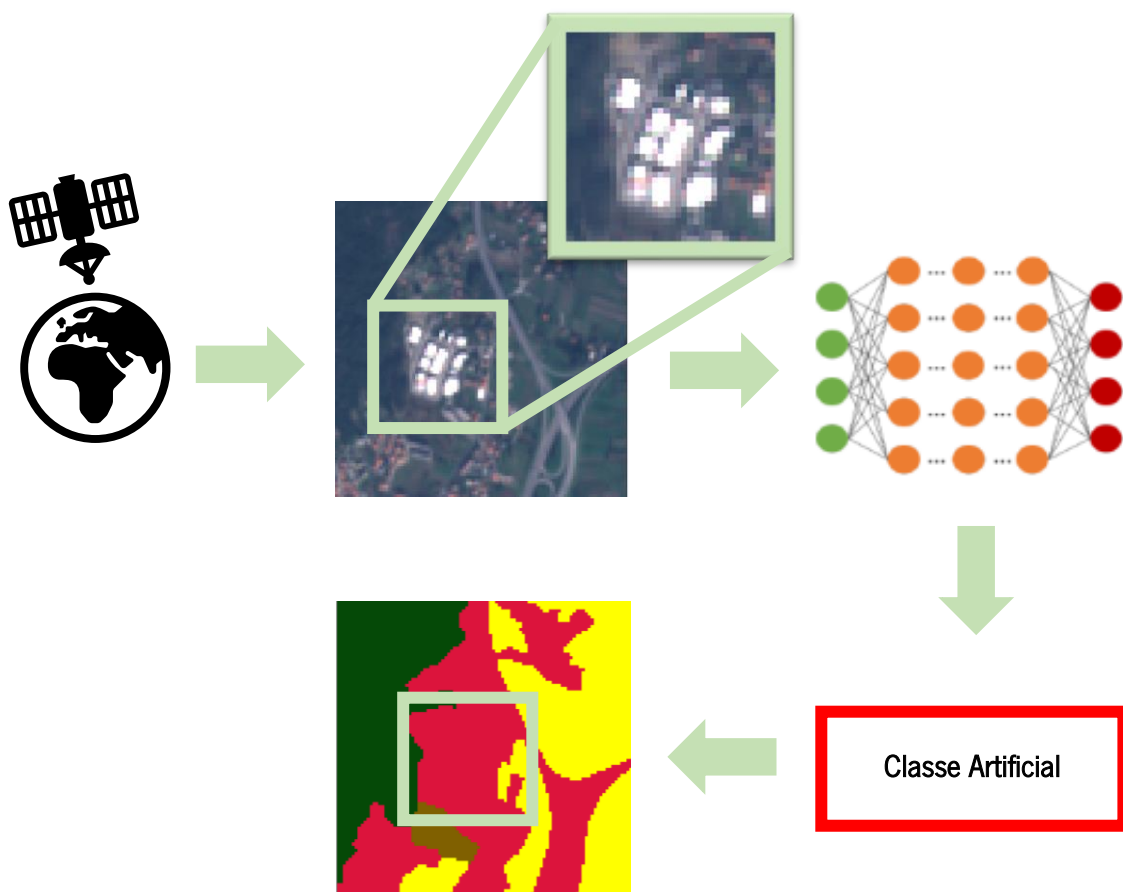


Figura 1 - Esquema do processo de classificação do coberto e uso do solo.

2. Base de dados e Classificador do Coberto e Uso do Solo

2.1. Dataset

Para se adquirir o *dataset*, começou-se por se obter um ficheiro GeoJSON que contém os limites da área a ser estudada. Com o objetivo de reduzir os recursos computacionais necessários, foi utilizada uma biblioteca do *Python Sentinel Hub* para dividir a área de interesse em várias *bounding boxes* de 5 por 5 Km.

É de notar, que para se fazer download deste *dataset*, tem que se automaticamente ter conta no [Sentinel Hub](#).

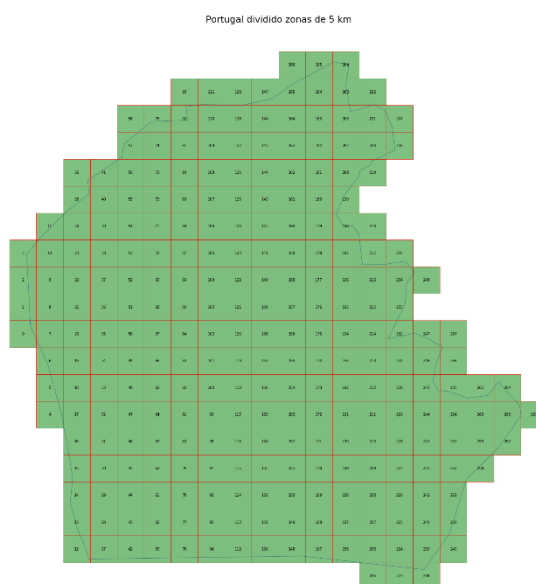


Figura 2 - Região Minho de Portugal dividida em bboxes de 5x5 km.

Através dos recursos fornecidos pelo *eo-learn*, é criado um *workflow*, que tem como primeira tarefa transferir, para cada *bounding box*, as imagens de satélite a partir do *Sentinel Hub* que estejam dentro de um determinado intervalo de tempo escolhido. Para além disso, esse *workflow* também cria EOPatches e preenche-os com os seguintes dados:

- Uma lista L1C personalizada de bandas, que correspondem aos comprimentos de onda [B,G,R,NIR,SWIR1,SWIR2]
- Uma máscara de nuvem, ou seja, uma máscara que deteta as nuvens na fotografia.
- Informações calculadas sobre NDVI, NDWI e NDBI, (índice de vegetação, índice de diferença normalizada da água, índice acumulado de diferença normalizada), utilizados para discriminar água, vegetação irrigada e solo exposto.

- Uma máscara de validade, com base nos dados adquiridos do *Sentinel Hub* e na cobertura da nuvem. O pixel só é válido caso:
 - A variável IS_DATA==True.
 - CLOUD_MASK == 0 (1 indica que tem nuvens, 255 indica que NO_DATA e 0 que tem dados, mas não tem nuvens).

O *workflow* definido anteriormente também executa uma tarefa de rasterização para ser possível associar a classificação do solo Português disponibilizada pela Carta de Uso e Ocupação do Solo de Portugal Continental (COS) às imagens transferidas através do *Sentinel Hub*.

```
workflow = LinearWorkflow(  
    add_data,  
    ndvi,  
    ndwi,  
    ndbi,  
    add_sh_valmask,  
    rasterization_task,  
    save  
)
```

Uma EOPatch representa toda a estrutura para cada *bounding box* e essa estrutura está representada da seguinte forma:

```
EOPatch(data: {  
    BANDS: numpy.ndarray(shape=(14, 500, 500, 6), dtype=float32)  
    CLP: numpy.ndarray(shape=(14, 500, 500, 1), dtype=uint8)  
    NDBI: numpy.ndarray(shape=(14, 500, 500, 1), dtype=float32)  
    NDVI: numpy.ndarray(shape=(14, 500, 500, 1), dtype=float32)  
    NDWI: numpy.ndarray(shape=(14, 500, 500, 1), dtype=float32)  
})  
  
mask: {  
    CLM: numpy.ndarray(shape=(14, 500, 500, 1), dtype=uint8)  
    IS_DATA: numpy.ndarray(shape=(14, 500, 500, 1), dtype=uint8)  
    IS_VALID: numpy.ndarray(shape=(14, 500, 500, 1), dtype=bool)  
}  
  
scalar: {}  
label: {}  
vector: {}  
data_timeless: {}  
mask_timeless: {  
    LULC: numpy.ndarray(shape=(500, 500, 1), dtype=uint8)  
    VALID_COUNT: numpy.ndarray(shape=(500, 500, 1), dtype=int64)  
}  
  
scalar_timeless: {}  
label_timeless: {}  
vector_timeless: {}  
meta_info: {    maxcc: 0.8    service_type: 'processing'    size_x: 500    size_y: 500    time_difference: datetime.timedelta(days=5)    time_interval: ('2018-01-01T00:00:00', '2018-12-31T23:59:59')  
}  
bbox: BBox(((515000.0, 4615000.0), (520000.0, 4620000.0)), crs=CRS('32629'))  
timestamp: [datetime.datetime(2018, 1, 8, 11, 34, 43), ..., datetime.datetime(2018, 12, 26, 11, 29, 38)], length=14)
```


2.2. Preparação do *Dataset*

Após a transferência das imagens de satélite, foi criado outro workflow para preparar os dados para o treino.

```
workflow = LinearWorkflow(  
    load,  
    concatenate,  
    filter_task,  
    linear_interp,  
    erosion,  
    spatial_sampling,  
    save  
)
```

Primeiramente fez-se a concatenação das estruturas BANDS, NDVI, NDWI, NDBI numa só estrutura (isto vai ficar num só ficheiro designado de FEATURES).

Foram removidas as imagens consideradas demasiado nubladas, ou seja, foi escolhido manter apenas os períodos de tempo em que a imagem tem pelo menos 80% de cobertura limpa.

Durante a aquisição das imagens satélites, é comum encontrar secções sem dados devido a, por exemplo, imagens removidas por condições atmosféricas. De modo a resolver este problema, foi feita uma interpolação temporal para 6 datas com um espaçamento de 2 meses com o objetivo de preencher os espaços em branco e obter as características de cada pixel em diferentes momentos do ano.

Também foi realizada erosão ao mapa com a classificação de referência para remover artefactos com largura de 1 pixel e remover contornos entre os polígonos de diferentes classes.

Por último, foi seleccionada uma amostra aleatória de pixéis que vão ser usados como *dataset* para alguns dos modelos utilizados (Ficheiro ALL_FEATURES_SAMPLED).

2.3. Classificador COS

Como já referido anteriormente escolheu-se o COS para classificar o solo Português. O COS (Carta de Uso e Ocupação do Solo de Portugal Continente) é uma cartografia temática que pretende classificar a ocupação/uso do solo no território de Portugal continental.

Optou-se por escolher este classificador ao CORINE, por causa do número de classes base que este apresenta. O nível mais básico da hierarquia de classes do CORINE tem apenas 5 classes e o objetivo era trabalhar com mais. Poderia-se utilizar o segundo nível de hierarquia do CORINE, mas esse já tinha 15 classes, que já eram demasiadas classes. O COS tem 9 classes no primeiro nível de hierarquia que na nossa opinião é um bom número de classes. de classes.

As classes definidas nesta cartografia são:

- Territórios artificializados
- Agricultura
- Pastagens
- Sistemas agroflorestais
- Florestas
- Matos
- Espaços descobertos ou vegetação esparsa
- Zonas húmidas
- Corpos de água

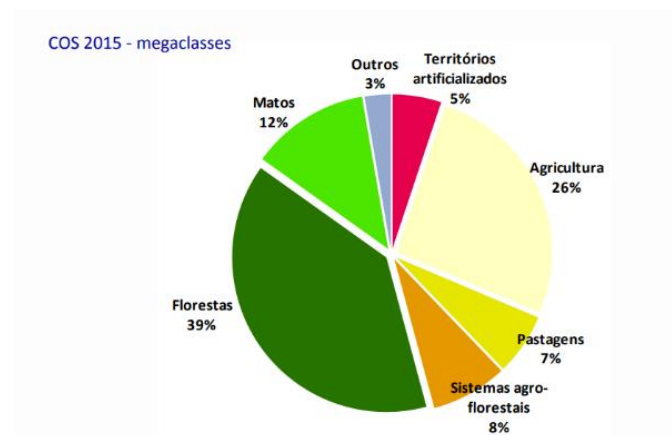


Figura 3 - Demonstração de cada classe no COS 2015.

Os **territórios artificializados** correspondem às superfícies de território destinadas a atividades de intervenção humana que inclui áreas de tecido urbano, industriais, comerciais, de serviços, jardins ou parques urbanos, equipamentos culturais e de lazer e as redes rodoviárias e ferroviárias.

A **Agricultura** é a superfície que é usado para o pratico dessa e as **pastagens** é o terreno onde há vegetação utilizada para a alimentação do gado.

Os **Sistemas Agroflorestais** é uma área que reúne as culturas de importância em consórcio com a floresta.

A **Floresta** é a zona com alta densidade de árvores e os **matos** são bastantes semelhantes às florestas, mas baseado em vegetação de menor porte como arbustos.

Os **espaços descobertos** ou vegetação esparsa que são do género de desertos com árvores ou arbustos bastante dispersados.

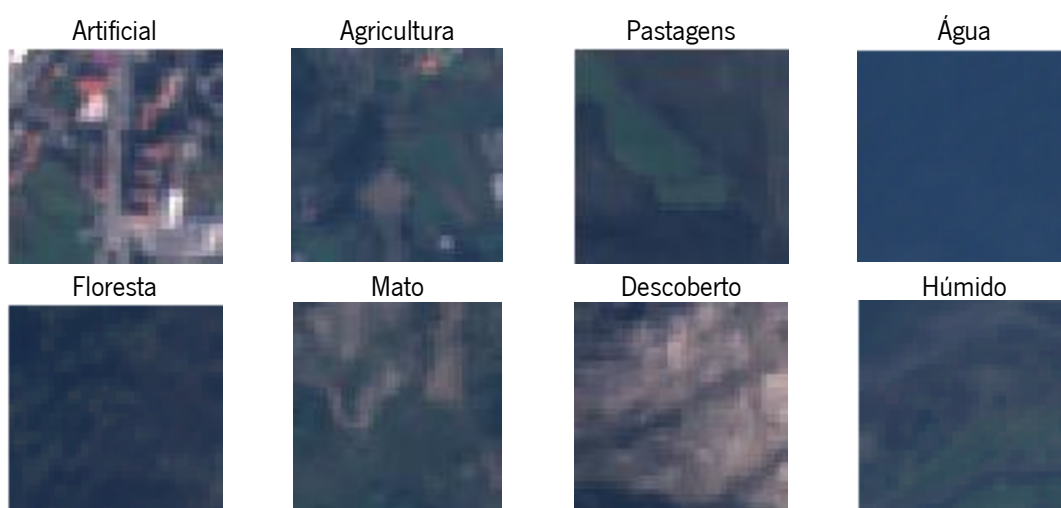
As **zonas húmidas** são ambientes intermédios entre os ecossistemas aquáticos e os ecossistemas terrestres, como zonas de pântanos ou salinas.

Finalmente a última classe **corpos de água** são praticamente todas as zonas aquáticas, oceanos, mares, rios, lagos, lagoas e lagunas.

Megaclassas de ocupação/uso do solo	COS21995		COS2007		COS2010		COS2015	
	ha	%	ha	%	ha	%	ha	%
Territórios artificializados	340038	3,8	434744	4,9	448128	5,0	455494	5,1
Agricultura	2510424	28,2	2320315	26,0	2326745	26,1	2340486	26,3
Pastagens	646262	7,3	602215	6,8	588205	6,6	575197	6,5
Sistemas agro-florestais	770001	8,6	719170	8,1	716468	8,0	712481	8,0
Florestas	3286448	36,9	3473509	39,0	3469742	38,9	3478646	39,0
Matos	1150109	12,9	1123525	12,6	1122100	12,6	1105702	12,4
Espaços descobertos ou com vegetação esparsa	61266	0,7	63592	0,7	63736	0,7	63717	0,7
Zonas húmidas	26141	0,3	26348	0,3	26344	0,3	26346	0,3
Corpos de água	119531	1,3	146801	1,6	148750	1,7	152151	1,7
Total Geral	8910220	100	8910220	100	8910220	100	8910220	100

Figura 4 - Tabela que demonstra o número de amostras de cada classificação em cada versão do COS para cada ano.

As imagens abaixo mostram um exemplo de cada classe vista de satélite.



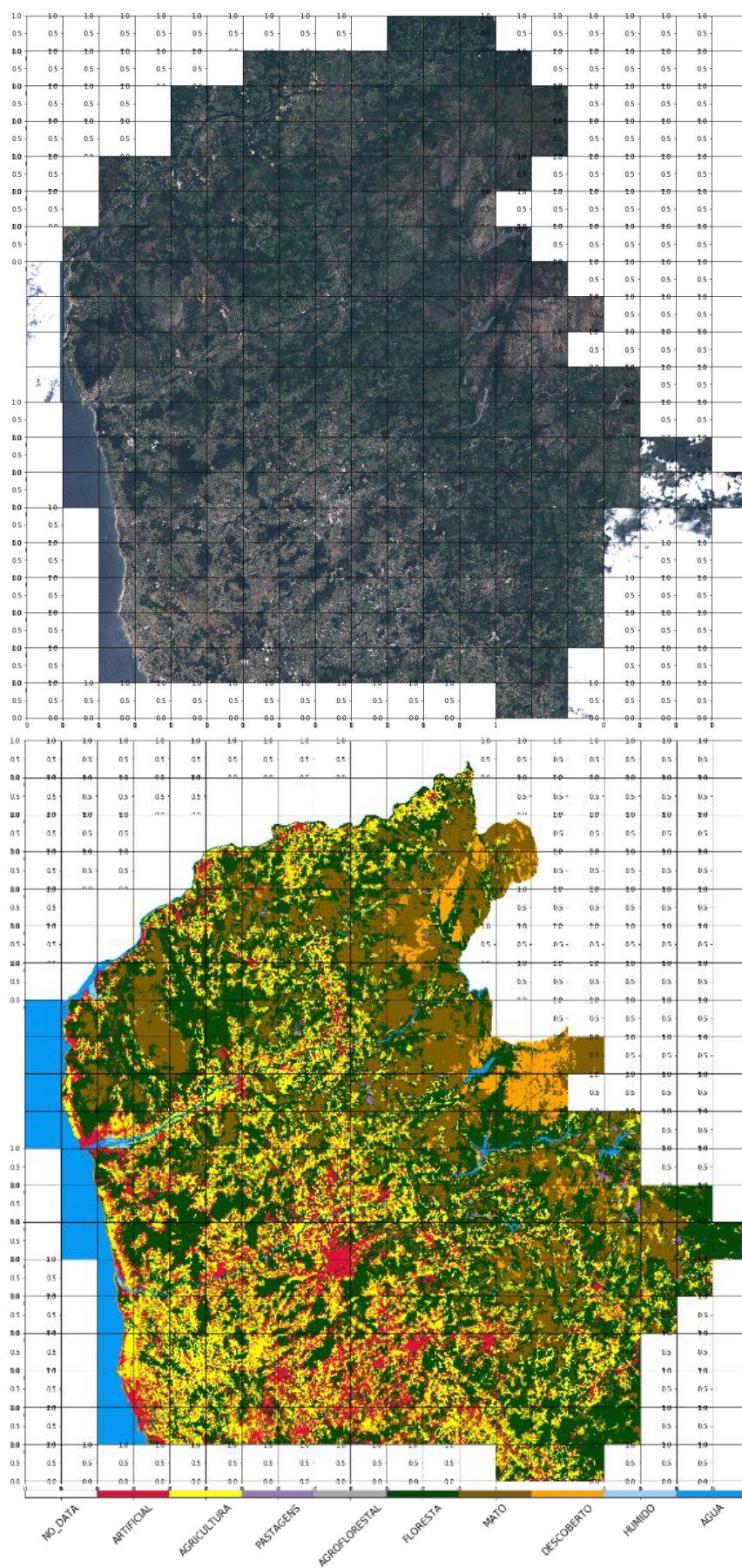


Figura 5 - Resultado da classificação pelo COS no Minho de Portugal.

3. Técnicas de Aprendizagem Automática

De forma a se procurar obter o melhor programa para classificar o coberto e solo, houve a necessidade de arranjar mais do que uma forma de se produzir *Machine Learning*.

3.1. Classificador LGBM

O primeiro método escolhido para a aprendizagem da máquina foi um classificador LGBM desenvolvido pela Microsoft. A LightGBM é uma estrutura de aumento de gradiente que usa algoritmos de aprendizagem baseados numa árvore. Segundo a sua documentação, este foi projetado para ser distribuído e eficiente com as seguintes características:

- Maior velocidade de treino e maior eficiência;
- Menor uso de memória;
- Melhor precisão;
- Suporte ao aprendizado paralelo e GPU;
- Capaz de lidar com dados em grande escala.

Para treinar este modelo foram utilizadas as amostras recolhidas e guardadas anteriormente. As amostras guardadas contêm, por pixel, 9 características para cada uma das 6 datas para as quais foi efetuada interpolação, assim como a sua classificação segundo o COS. Sendo assim, por pixel temos um total de 54 características que irão treinar o modelo para classificação do solo.

```
# Set up training classes
labels_unique =
np.unique(labels_train)

# Set up the model
model = lgb.LGBMClassifier(
    objective='multiclass',
    num_class=len(labels_unique),
    metric='multi_logloss',
    random_state=42,
    class_weight='balanced'
)

# train the model
model.fit(features_train,
labels_train)
```

3.2. Rede Neuronal

Uma rede neuronal consiste numa estrutura colecionista, na qual o processo se encontra distribuído por um grande número de pequenas unidades densamente interligadas. O objetivo da criação e experimentação desta rede, foi evitar se fazer modificações em relação ao classificador anterior, o input continua a ser em pixels, apenas vão ser treinados agora numa rede neuronal.

Para treinar a rede neuronal são selecionadas até 30 000 amostras de cada uma das classificações de solo existentes para uma das datas interpoladas que não contenha dados nulos. Essas amostras e as suas respetivas classificações são usadas para criar o *dataset* de treino em

que 20% vai ser usado para validação das épocas. As *batches* deste *dataset* têm um tamanho de 512.

A rede feita de raiz para este problema foi a seguinte:

```
model = Sequential()

model.add(Dense(256,
                input_shape=(channels,),
                activation='relu'))

model.add(BatchNormalization())

model.add(Dense(256, activation='relu') )
model.add(Dense(128, activation='relu') )
model.add(Dense(64, activation='relu') )
model.add(BatchNormalization())

model.add(Dense(classCount, activation='softmax'))

opt = Adam(lr=0.001)
model.compile(optimizer = opt,
              loss='categorical_crossentropy',
              metrics=[ tfmetrics.categorical_accuracy])
```

Na **Figura 6** apresentam-se os gráficos da evolução da *accuracy* e da *loss* durante o treino da rede.

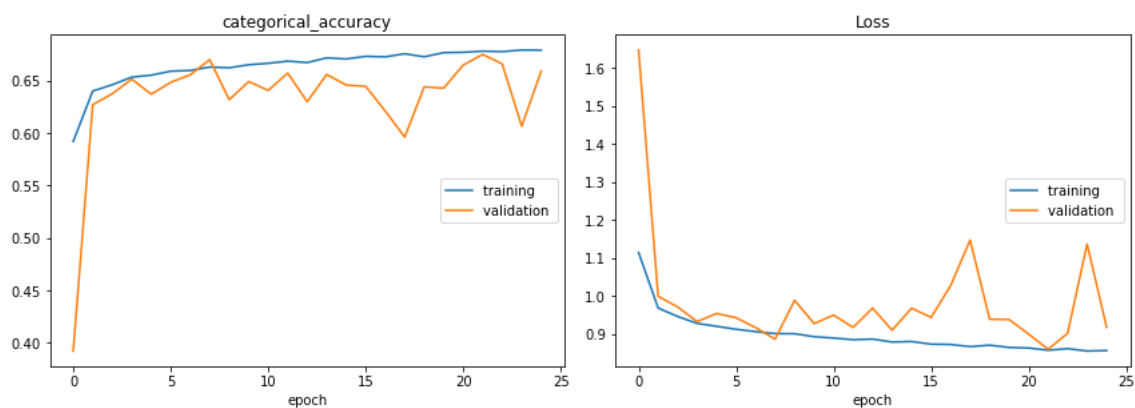


Figura 6 - Gráficos da evolução da accuracy e loss ao longo das épocas para a rede neuronal.

3.3. CNN

Uma rede neuronal convolucional é uma classe de redes neuronal do tipo *feed-forward*, que vem sendo bastante aplicada e com sucesso no processamento e análise de imagens digitais. Para este projeto escolheu-se experimentar a famosa rede de transferência ResNet50, ResNet34 e uma rede feita manualmente, que foi sendo modificada e melhorada ao longo do processo de forma a se obter bons resultados.

Para se treinar este *dataset* neste tipo de redes neuronais, teve que se adaptar e modificar o input da rede, que no classificador LGBM e na rede neuronal anterior era um pixel, e agora passa a ser uma imagem de 32x32 pixels que se podem extrair dos *patches* guardados com as imagens de satélite.

Para se extrair, primeiramente identificam-se as localizações de todos os pixels válidos para treino e organizam-se estas localizações por classe. Para isso cria-se o dicionário `class_locations`, que contém uma lista com as localizações dos pixels para cada classe existente. As localizações dos pixels são definidas são representadas por tuplos (pid,x,y), onde pid identifica o *patch* em que o pixel se encontra e x e y representam as coordenadas do pixel dentro do *patch*.

De seguida é feita uma amostragem dessas listas para se obter até 3000 pixels por classe que, juntamente com os pixels vizinhos destes, serão utilizados para se obter imagens de tamanho 32x32 pixels. Destas imagens são escolhidas 20% para o teste das redes e as restantes imagens são utilizadas para o treino (80%) e validação (20%) das redes.

3.3.1. ResNet50

Como já referido anteriormente, a Resnet50 é uma rede de transferência, isto é, uma rede pré-treinada. Um modelo pré-treinado, como o próprio nome indica, foi treinado anteriormente num conjunto de dados e contém pesos e preconceitos que representam os recursos de qualquer conjunto de dados em que foi treinado. Os recursos aprendidos geralmente são transferíveis para dados diferentes. [1] Uma das maiores vantagens de se utilizar uma rede pré-treinada é economizar tempo. Pode encontrar a arquitetura desta rede [aqui](#).

```
base_model = tf.keras.applications.ResNet50V2(  
    include_top=False,  
    weights=None,  
    input_tensor=None,  
    input_shape=(IMAGE_SIZE, IMAGE_SIZE, 1*9),  
    pooling=None,  
)
```

Mas como se acrescentou camadas e fez se modificação na rede ResNet50, não se utilizou os pesos já pré-treinados pela rede, ou seja, foi como se tivesse feito uma rede de raiz. Para além das camadas já definidas na rede ResNet50, foram acrescentadas as seguintes:

```
x = base_model.output
x = keras.layers.GlobalAveragePooling2D()(x)
x = keras.layers.Dense(1024)(x)
x = keras.layers.LeakyReLU()(x)
x = keras.layers.Dense(256)(x)
x = keras.layers.LeakyReLU()(x)
x = keras.layers.Dense(10, activation='softmax')(x)

model = keras.models.Model(inputs=base_model.input,
                           outputs=x)
```

Esta rede foi treinada durante 100 épocas utilizando o otimizador *Adam*, uma *learning rate* de 0.0001 e um tamanho de *batches* de 512.

Na **Figura 7** observam-se os gráficos da evolução da *accuracy* e da *loss* durante o treino desta rede. Nesta observa-se grandes diferenças entre os valores obtidos para no treino e os valores obtidos durante a validação e uma *accuracy* perto de 100% durante o treino indicando que a rede esteja a fazer *overfitting* ao problema. Este *overfitting* pode-se dever à elevada complexidade da rede para este problema ou uma amostragem demasiado pequena, no entanto, também foi realizada uma amostragem com tamanho de 12500 amostras por classe resultando no mesmo problema, pelo que de seguida experimentam-se redes mais simples.

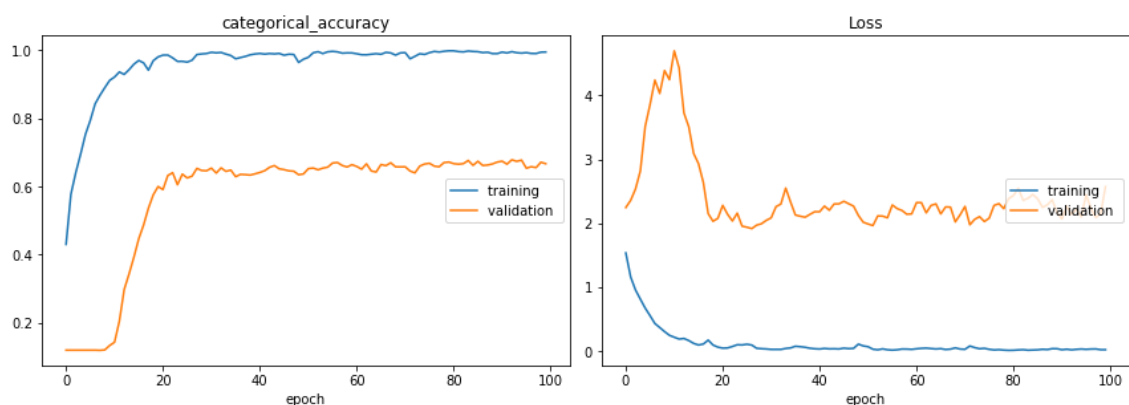


Figura 7 - Gráficos da evolução da *accuracy* e *loss* da ResNet50 ao longo de 100 épocas.

3.3.2. Resnet34

A rede ResNet34 testada utiliza as mesmas camadas finais, épocas, *learning rate* e tamanho de *batches* que a rede anterior. Na **Figura 8** observam-se os gráficos obtidos para a accuracy e a loss durante o treino desta rede. Nestes gráficos encontram-se os mesmos problemas que se encontravam na rede anterior, mas com uma accuracy final na validação ligeiramente superior (69.81%) que a rede anterior (66.63%).

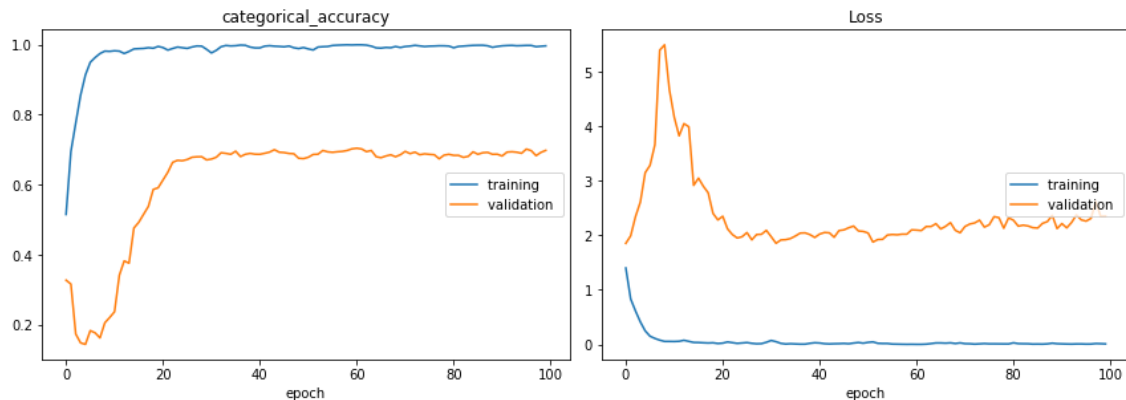


Figura 8 - Gráficos da evolução da *accuracy* e *loss* da ResNet34 ao longo de 100 épocas.

3.3.3. Rede convolucional de raiz

Para além da utilização das redes convolucionais ResNet, também foi criada uma rede convolucional de raiz. A rede apresenta a seguinte estrutura:

```
model = Sequential()
model.add(Conv2D(32, (2, 2), padding='same',
                 input_shape=(imgSize, imgSize, channels),
                 activation='relu'))
model.add(Conv2D(32, (2, 2), padding='same', activation='relu') )
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.4))
model.add(Conv2D(64, (3, 3), padding='same', activation='relu') )
model.add(Dropout(0.4))
model.add(Conv2D(64, (3, 3), padding='same', activation='relu') )
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.3))
model.add(Conv2D(128, (3, 3), padding='same', activation='relu') )
model.add(Dropout(0.3))
model.add(Conv2D(128, (3, 3), padding='same', activation='relu') )
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(classCount, activation='softmax'))
opt = Adam(lr=0.001)
model.compile(optimizer = opt, loss='categorical_crossentropy',
              metrics=[ 'categorical_accuracy' ])
```

Esta rede foi treinada durante 100 épocas com o optimizador Adam, uma *learning rate* de 0.001 e um tamanho de *batches* de 512.

Na **Figura 9** apresentam-se os gráficos da evolução da *accuracy* e da *loss* durante o treino desta rede. Nestes observa-se que a *accuracy* tem vindo sempre a melhorar, pelo que esta rede provavelmente deveria ter sido treinada durante mais épocas para atingir melhores resultados. Nesta rede mais simples também já não observamos o *overfitting* que ocorria nas redes anteriores.

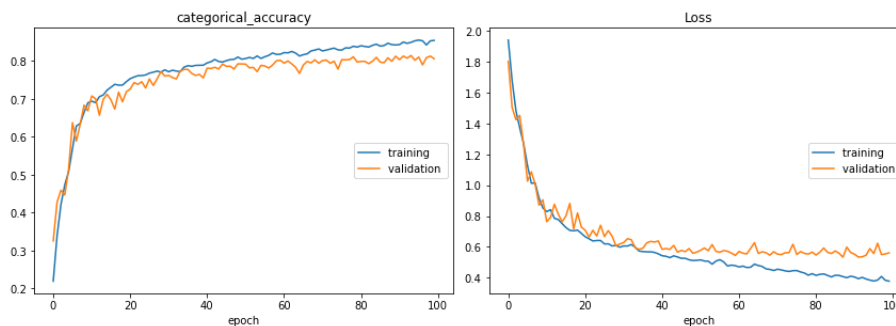


Figura 9 - Gráficos da evolução da *accuracy* e *loss* ao longo das épocas para a rede convolucional de raiz.

4. Validação

Terminada a aprendizagem da classificação do solo português, é necessário validar os resultados obtidos pelas redes neuronais e pelo classificador LGBM.

A validação foi calculada usando as seguintes métricas:

- **Accuracy:** é a medida de desempenho mais intuitiva e é simplesmente uma proporção das observações previstas corretamente com o total de observações.
- **Precision:** é a razão entre as observações positivas previstas corretamente e o total de observações positivas previstas.
- **Recall:** calcula quantos positivos o modelo captura (Verdadeiro Positivo), ou seja, é a proporção entre as observações positivas previstas corretamente e todas as observações de uma classe.
- **F₁ score:** é a média ponderada de *Precision* e *Recall*, essa pontuação leva a cabo tanto os falsos positivos quanto os falsos negativos.

Adicionalmente, foi criada a matriz de confusão que mostra as categorias que foram corretamente e incorretamente previstas.

A matriz confusão é uma boa medição de desempenho para o problema de classificação, é uma tabela que mostra as frequências de classificação para cada classe do modelo. Ela vai mostrar:

- Verdadeiro positivo (*true positive* - TP): ocorre quando no conjunto real, a classe que foi prevista corretamente. Foi previsto positivo e é verdadeiro.
- Falso positivo (*false positive* – FP) (erro tipo 1): ocorre quando no conjunto real, a classe foi prevista incorretamente. Foi previsto positivo e é falso.
- Verdadeiro negativo (*true negative* - TN): ocorre quando no conjunto real, a classe que não é a que se está a buscar, é classificada corretamente. Foi previsto negativo e é verdadeiro.
- Falso negativo (*false negative* – FN) (erro tipo 2): ocorre quando no conjunto real, a classe que não se está a buscar for prevista incorretamente. Foi previsto negativo e é falso.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

4.1. Classificador LGBM

O classificador LGBM teve um accuracy de 72.5% e F1-score de 75.4%. Na tabela abaixo estão representadas as métricas F1-score, Recall e Precision para cada classe.

Class	= F1	Recall	Precision
-			
* ARTIFICIAL	= 74.6	79.9	69.9
* AGRICULTURA	= 73.3	67.3	80.5
* PASTAGENS	= 11.1	72.6	6.0
* AGROFLORESTAL	= 4.4	54.5	2.3
* FLORESTA	= 81.7	73.9	91.3
* MATO	= 61.8	60.3	63.3
* DESCOBERTO	= 46.8	83.9	32.4
* HUMIDO	= 11.9	94.3	6.4
* AGUA	= 98.9	98.8	99.0

Calculou-se a matriz confusão e a sua transposta de forma a facilitar a análise dos resultados.

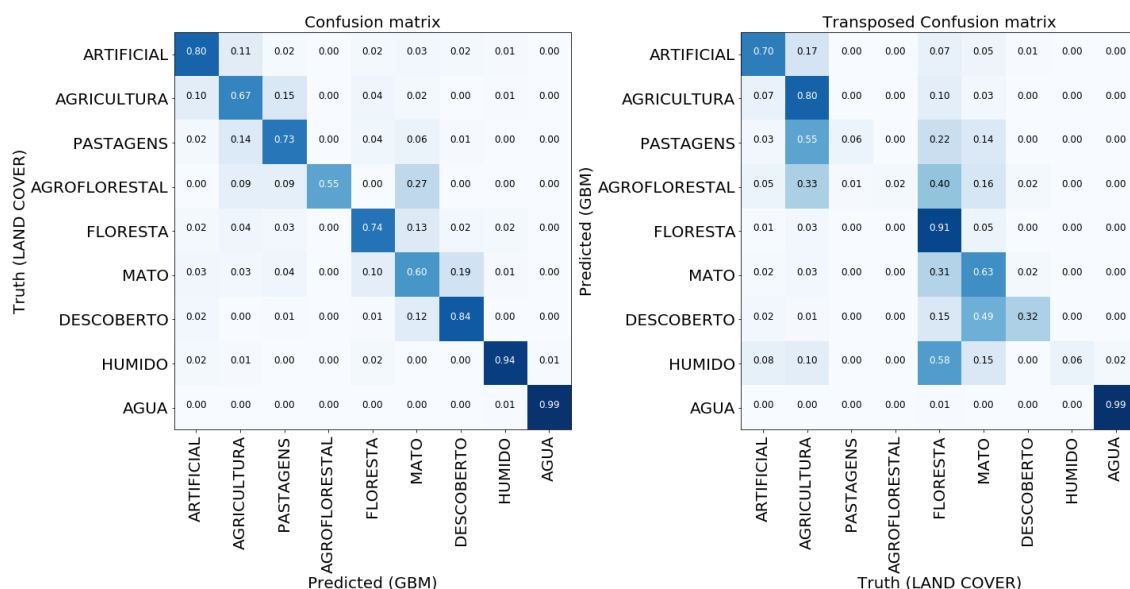


Figura 10 - Matriz confusão e a transposta do classificador LGBM.

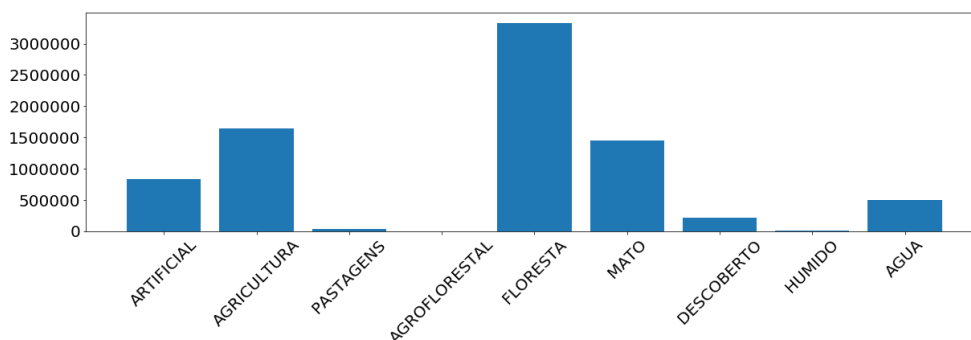


Figura 11 - Quantidade de amostras para cada classe que foram usadas no treino no Classificador LGBM.

4.2. Rede Neuronal – por pixel

A rede neuronal teve um accuracy de 65.1% e F1-score de 64.9%. Na tabela abaixo está representado o F1-score, Recall e Precision para cada classe.

Class	= F1	Recall	Precision
* ARTIFICIAL	= 70.5	63.2	79.7
* AGRICULTURA	= 50.3	45.4	56.5
* PASTAGENS	= 54.3	53.2	55.4
* AGROFLORESTAL	= 0.0	0.0	0.0
* FLORESTA	= 65.8	74.5	58.9
* MATO	= 41.6	41.8	41.4
* DESCOBERTO	= 70.0	77.5	63.8
* HUMIDO	= 74.1	71.0	77.4
* AGUA	= 97.7	97.7	97.7

Na figura 12 é possível visualizar a matriz confusão e a sua transposta.

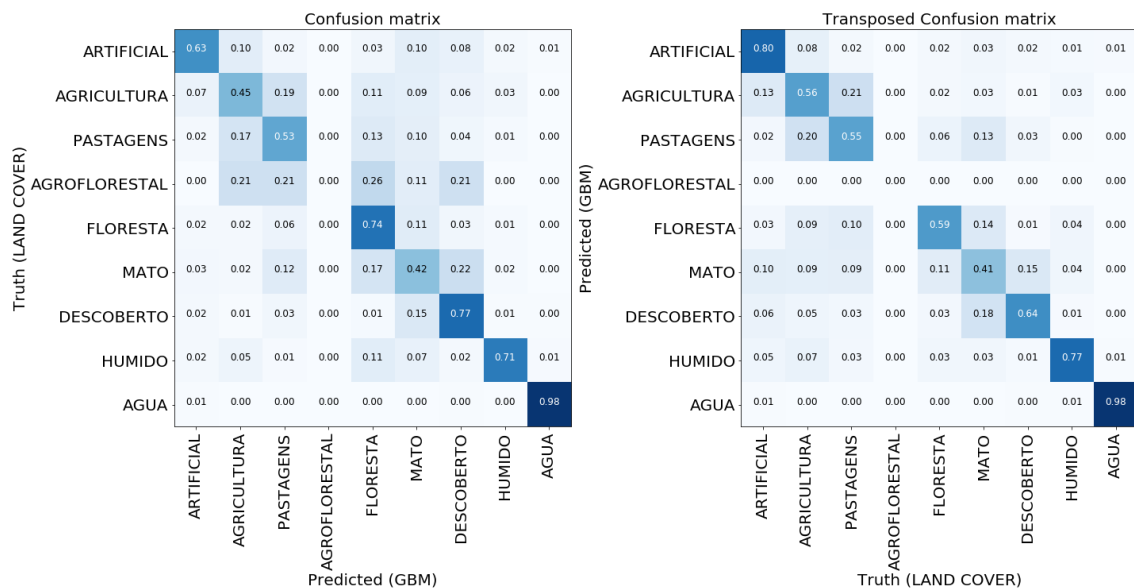


Figura 12 - Matriz confusão e a transposta da rede neuronal por pixel

Na figura 13 tem a quantidade de amostras que foram usadas para cada classe no treino.

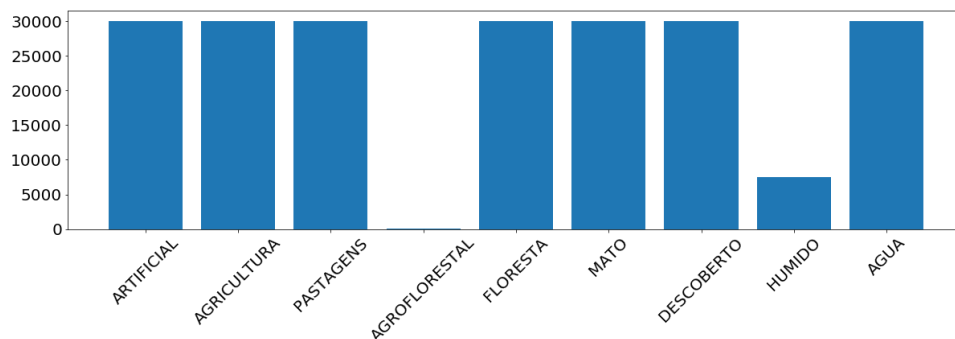


Figura 13 - Quantidade de amostras para cada classe que foram usadas no treino na Rede Neuronal por pixel.

4.3. CNN

4.3.1. ResNet50

A ResNet50 teve uma Accuracy geral de 66.2% e um F1-score também de 66.5%. Na tabela abaixo estão representadas as métricas F1-score, Recall e Precision para cada classe.

Class	= F1	Recall	Precision
* ARTIFICIAL	= 67.4	69.3	65.5
* AGRICULTURA	= 48.9	64.8	39.3
* PASTAGENS	= 45.8	38.6	56.4
* AGROFLORESTAL	= 96.3	94.5	98.1
* FLORESTA	= 59.6	50.3	72.9
* MATO	= 53.9	56.9	51.2
* DESCOBERTO	= 71.3	69.5	73.1
* HUMIDO	= 84.3	81.6	87.2
* AGUA	= 96.2	94.6	98.0

A matriz confusão e a sua transposta estão representadas na figura abaixo, respetivamente.

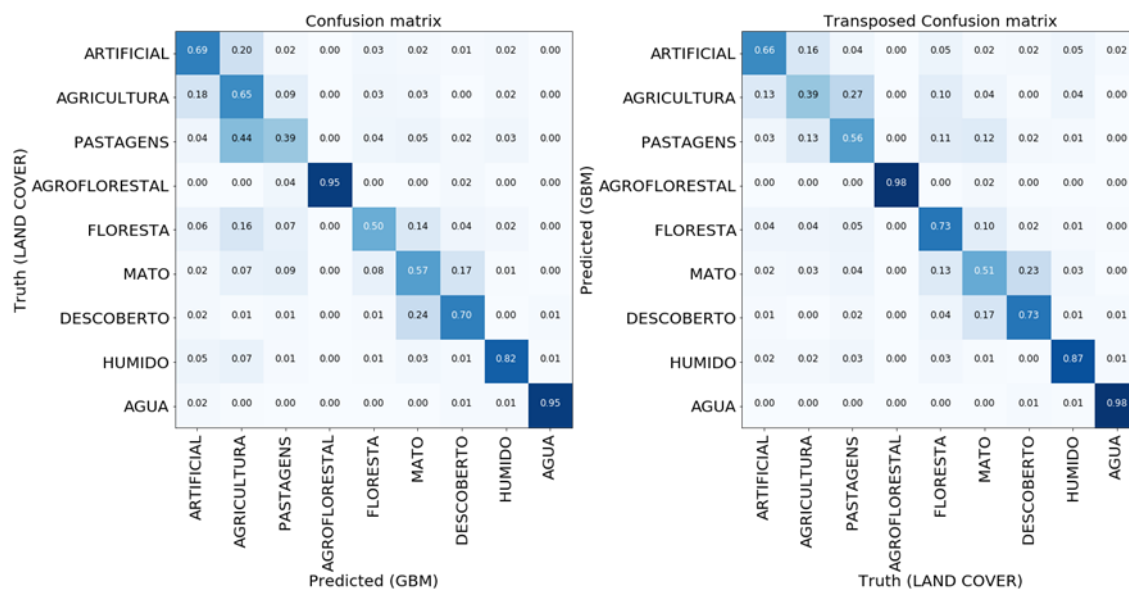


Figura 14 - Matriz confusão e transposta da ResNet50.

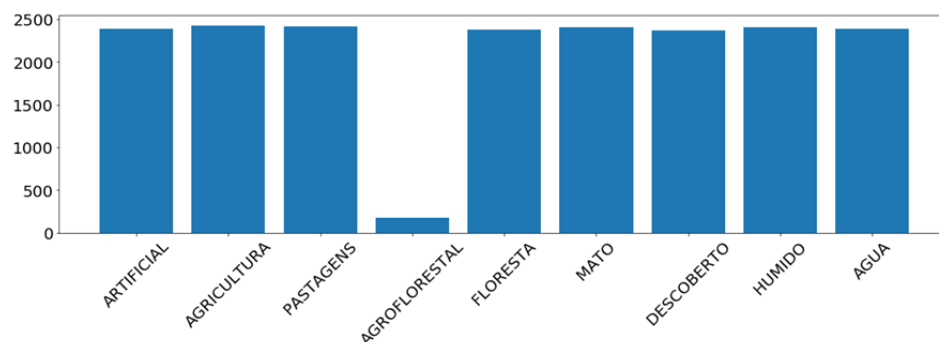


Figura 15 - Quantidade de amostras para cada classe que foram usadas no treino na ResNet50.

4.3.2. ResNet34

A Resnet34 teve uma accuracy de 70.3% e um F1-score 69.7%. Na tabela em baixo está representado F1-score, recall e precision para cada classe.

Class	=	F1		Recall		Precision
* ARTIFICIAL	=	71.3		80.7		63.8
* AGRICULTURA	=	51.0		51.3		50.7
* PASTAGENS	=	57.6		50.7		66.7
* AGROFLORESTAL	=	98.2		100.0		96.5
* FLORESTA	=	62.6		59.3		66.3
* MATO	=	48.5		44.1		54.0
* DESCOBERTO	=	73.2		79.6		67.7
* HUMIDO	=	92.3		96.1		88.8
* AGUA	=	96.2		94.9		97.6

Na seguinte figura está representado a matriz confusão e a sua transposta, respetivamente

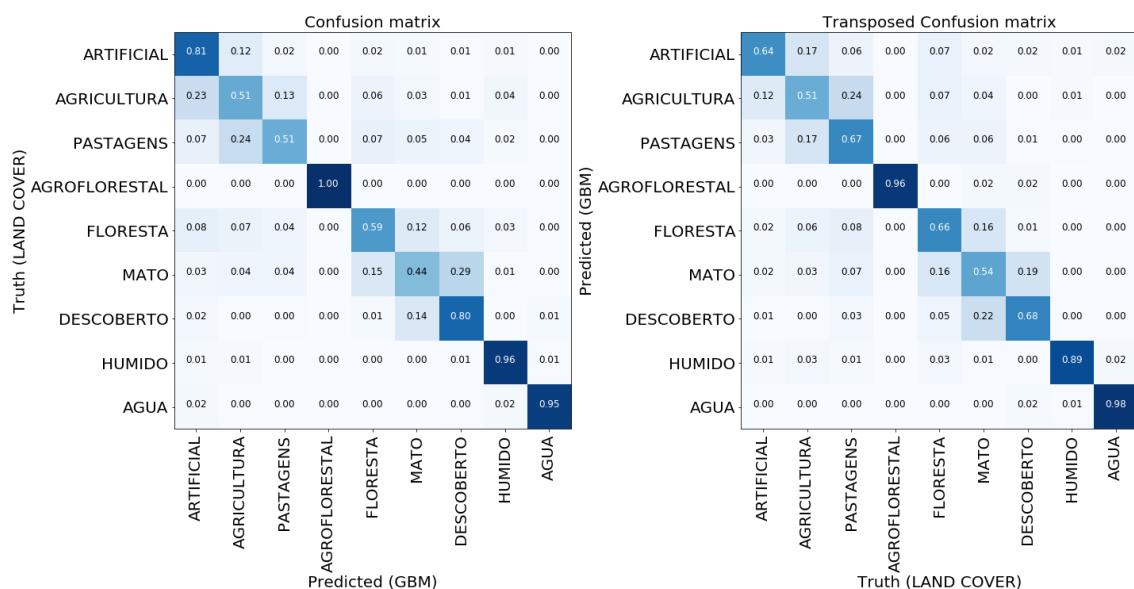


Figura 16 - Matriz confusão e transposta da ResNet34.

Na figura em baixo tem a quantidade amostras utilizadas para cada classe.

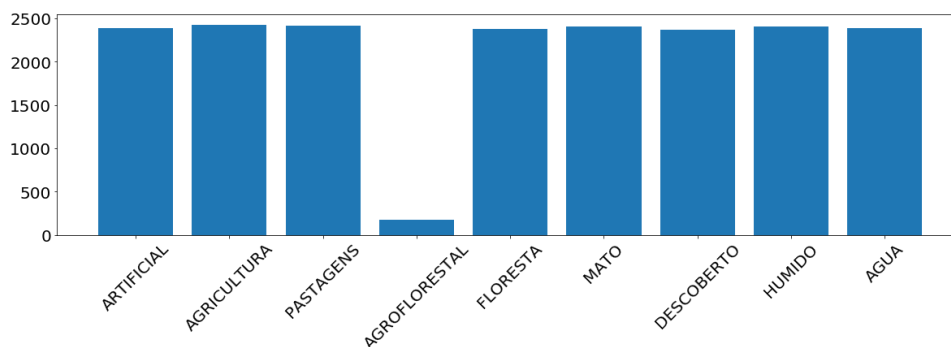


Figura 17 - Quantidade de amostras para cada classe que foram usadas no treino na ResNet34.

4.3.3. Rede convolucional de raiz

Esta rede teve uma Accuracy geral de 80.1% e um F1-score também de 79.9%. Na tabela abaixo estão representadas as métricas F1-score, Recall e para cada classe.

Class	= F1	Recall	Precision
* ARTIFICIAL	= 85.1	85.0	85.3
* AGRICULTURA	= 61.6	56.8	67.2
* PASTAGENS	= 76.3	82.6	71.0
* AGROFLORESTAL	= 100.0	100.0	100.0
* FLORESTA	= 71.0	65.9	77.0
* MATO	= 63.3	64.2	62.5
* DESCOBERTO	= 81.8	85.8	78.1
* HUMIDO	= 98.6	100.0	97.2
* AGUA	= 97.6	96.5	98.7

A matriz confusão e a sua transposta estão representadas na figura abaixo, respetivamente.

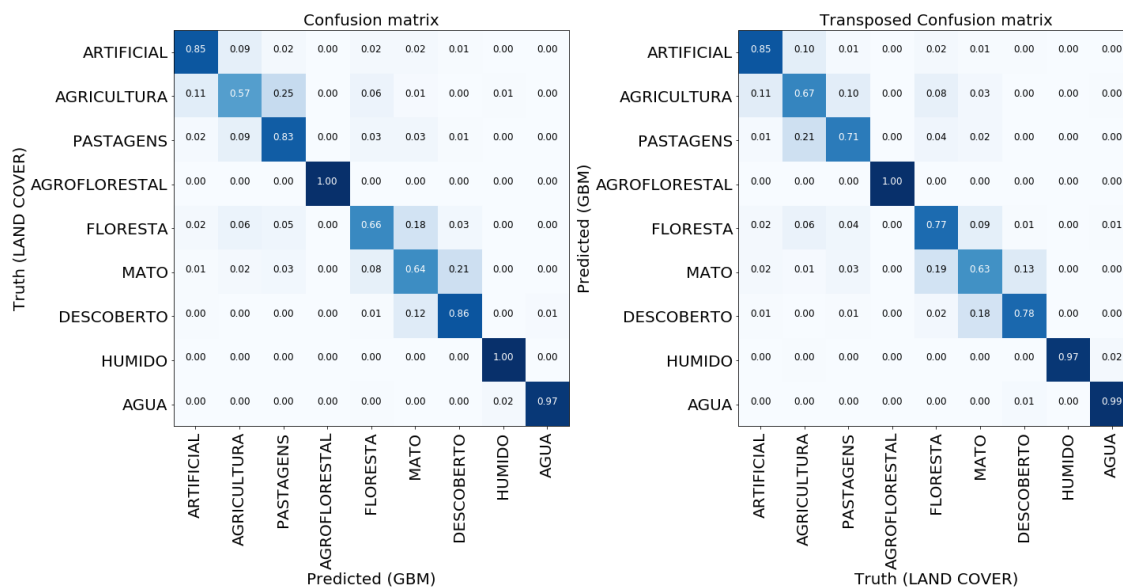


Figura 18 - Matriz confusão e transposta da CNN de raiz.

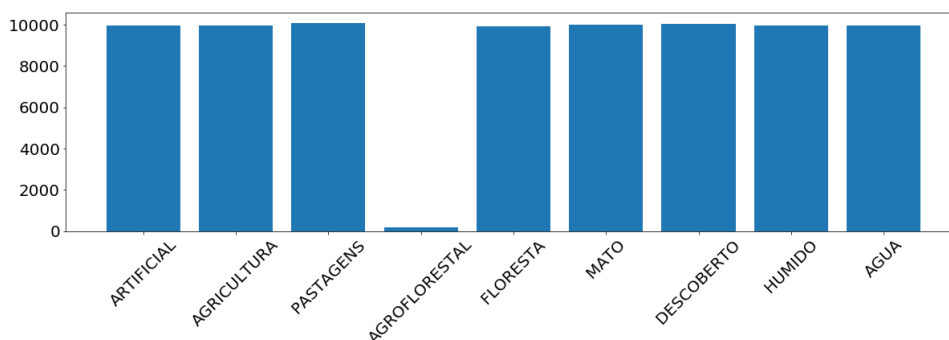


Figura 19 - Quantidade de amostras para cada classe que foram usadas no treino na CNNraiz.

5. Análise de resultados

Observando bem as métricas e matrizes de cada método de aprendizagem automática escolhido, resultantes da validação, verificou-se que o que teve melhores resultados foi a CNN de raiz com 84.9% de Accuracy, de seguida a ResNet50 com 81.4%, o classificador LGBM com 75.4%, e por último com piores resultados, a Rede Neuronal por pixel com 65.1% de Accuracy.

Através das matrizes confusões consegue-se entender que há um padrão de confusão pela aprendizagem dos modelos escolhidos, e que esse padrão vai sendo modificado ligeiramente caso o modelo tenha uma aprendizagem melhor.

Consegue-se observar que existe uma confusão normalmente entre as classes de **agricultura** e **pastagens**, e entre as classes **floresta** e **mato**. Achou-se que isso acontece porque essas classes são bastante semelhantes visualmente nas imagens de satélite. Outra classe que por vezes não conseguiu ser treinada em alguns modelos, nem tinha resultados bons de aprendizagem, foi a **Agroflorestal**, porque essa classe teve sempre menos amostras que as outras. Apesar destes pequenos problemas que até são bastantes comuns, conseguiu-se arranjar um modelo que minimizou estes erros e que apresentou melhores resultados.

5.1. Classificador LGBM

Na **figura 20** é apresentado a classificação do solo do Minho previsto pelo classificador LGBM. O resultado é bastante semelhante à classificação pelo COS exposta na figura 5.

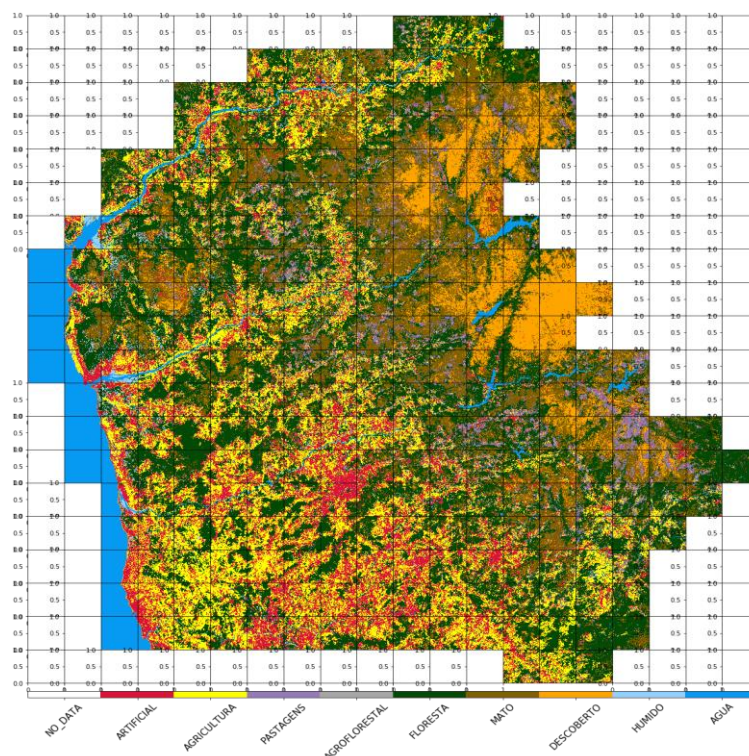


Figura 20 - Previsão da cobertura do solo pelo classificador LGBM

Na figura 21 foi escolhido um subconjunto de patches aleatoriamente, de forma a ser possível comparar e inspecionar uma parte da área de interesse de forma mais atenta e pormenorizada.

No canto superior esquerdo é mostrado o mapa com as classificações reais mapa obtidas pelo COS e no canto superior direito está a previsão feita pelo modelo treinado. Em baixo é apresentada a diferença entre as duas classificações e a imagem de satélite

Nessa figura, já é possível visualizar facilmente as diferenças da previsão do classificador em relação ao COS.

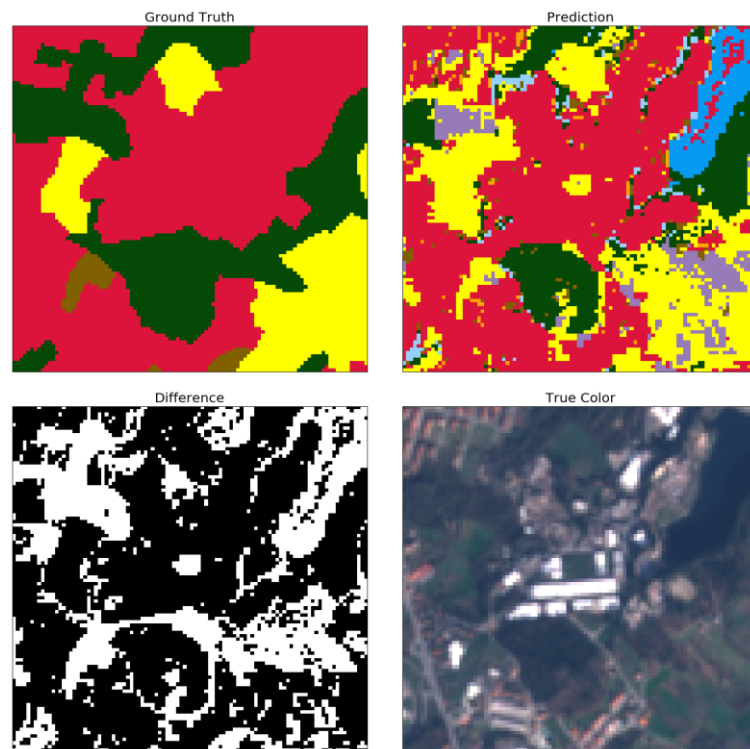


Figura 21 - Comparação da classificação do solo real com a classificação prevista pelo classificador LGBM

5.2. Rede Neuronal – por pixel

Tal como no classificador LGBM, é apresentado a previsão da classificação do solo do Minho pela Rede Neuronal por pixel.

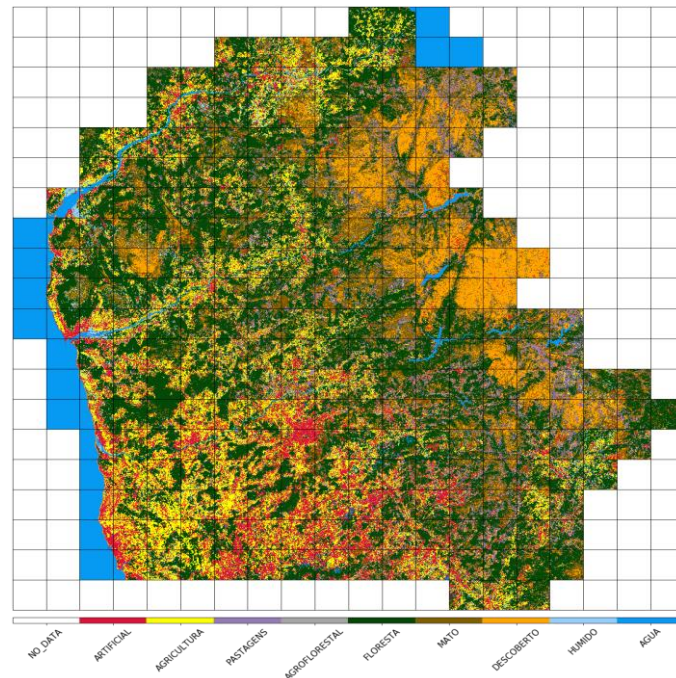


Figura 22 - Previsão da cobertura do solo pela rede neuronal por pixel

Na figura abaixo é possível comparar as classes do COS com a previsão da rede neuronal por pixel, de uma parte da área do Minho.

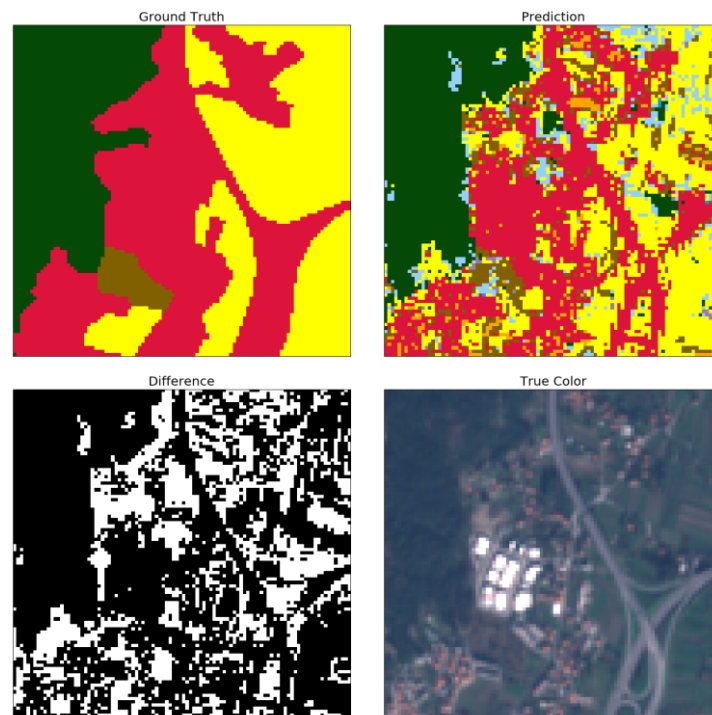


Figura 23 - Comparação da classificação do solo real com a classificação prevista pela rede neuronal por pixel

5.3. CNN

5.3.1. ResNet50

Na figura abaixo mostra a previsão da área do Minho pela rede ResNet50.

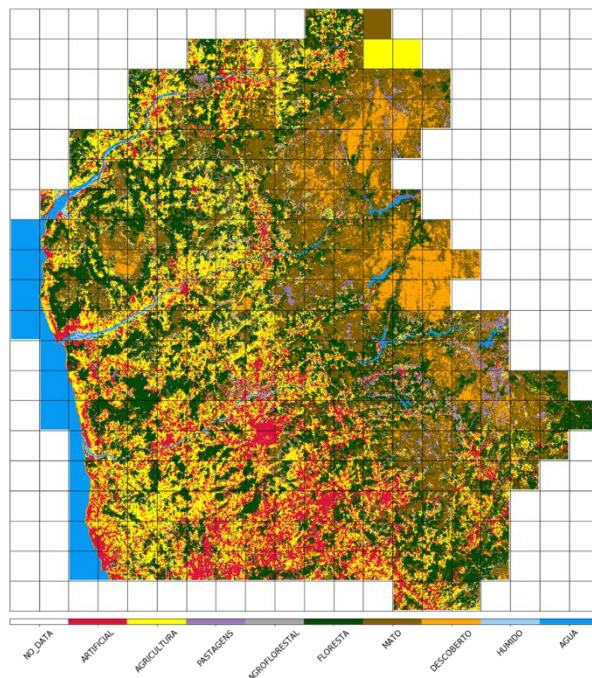


Figura 24 - Previsão da cobertura do solo pela rede ResNet50

De seguida é apresentado a comparação de uma pequena área da classificação feito pelo COS com a previsão da rede ResNet50.

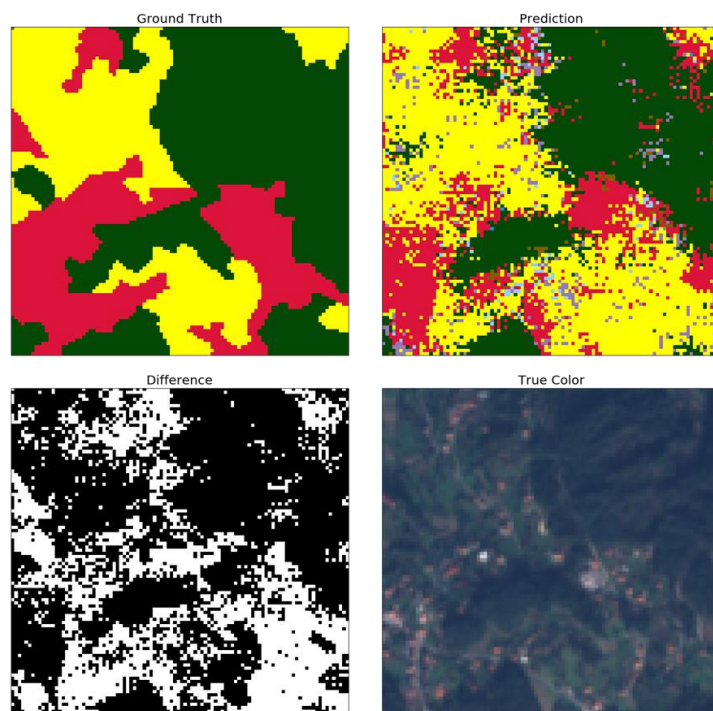


Figura 25 - Comparação da classificação do solo real com a classificação prevista pela rede ResNet50

5.3.2. ResNet34

Na figura em abaixo mostra a previsão da área do Minho pela rede ResNet34.

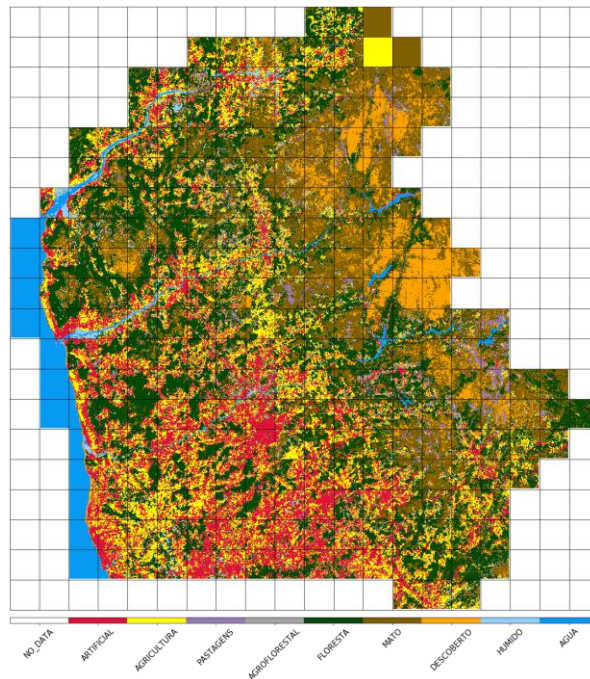


Figura 26 - Previsão da cobertura do solo pela rede ResNet34

Na seguinte figura é apresentado a comparação de uma pequena área da classificação feito pelo COS com a previsão da rede ResNet34.

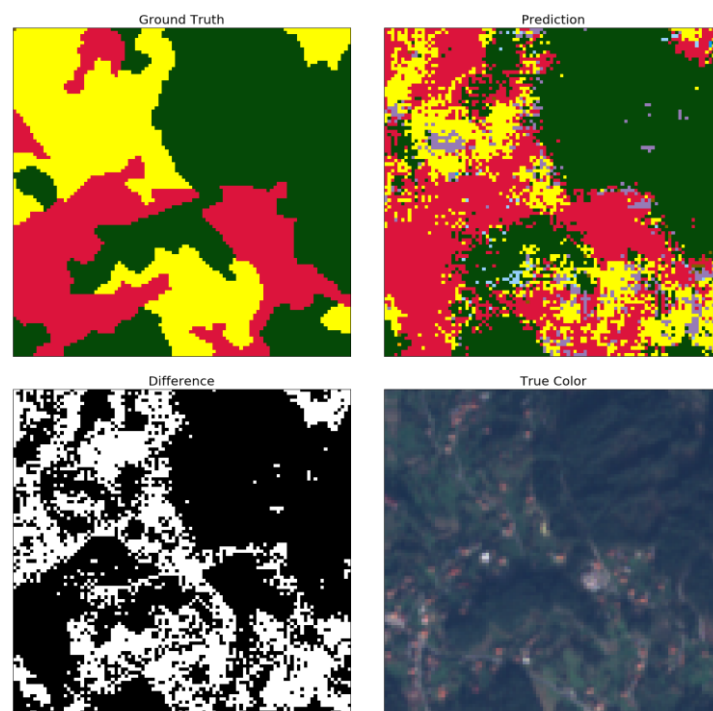


Figura 27 - Comparação da classificação do solo real com a classificação prevista pela rede ResNet34

5.3.3. Rede convolucional de raiz

Tal como foi feito para as outras redes e o classificador LGBM, é mostrado para a rede convolucional a previsão da área do Minho.

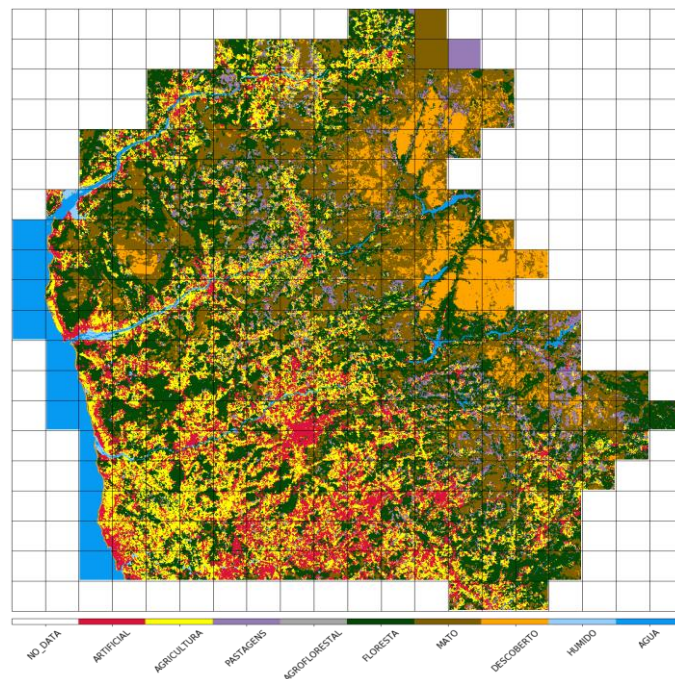


Figura 28 - Previsão da cobertura do solo pela rede convolucional de raiz

A figura em baixo tem a classificação de uma pequena área do Minho, a previsão feita pela rede convolucional de raiz, a diferença entre as duas imagens e, por último, a imagem de satélite.

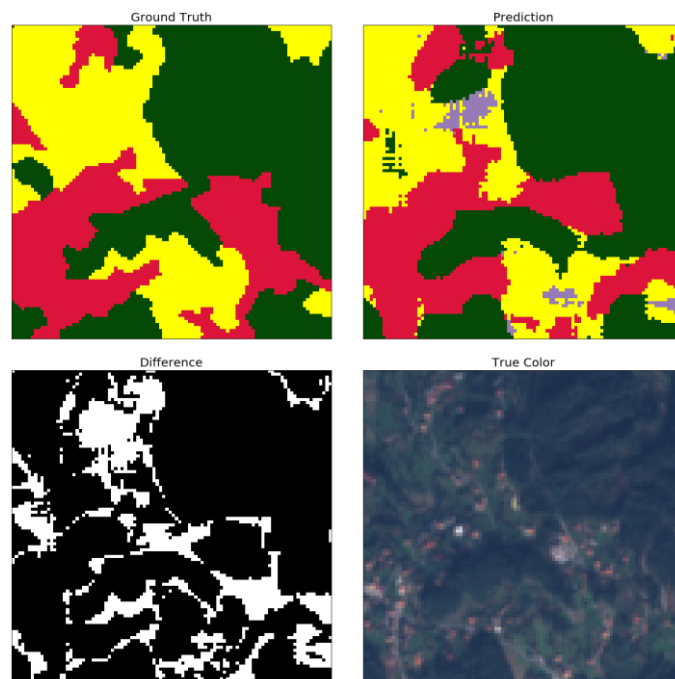
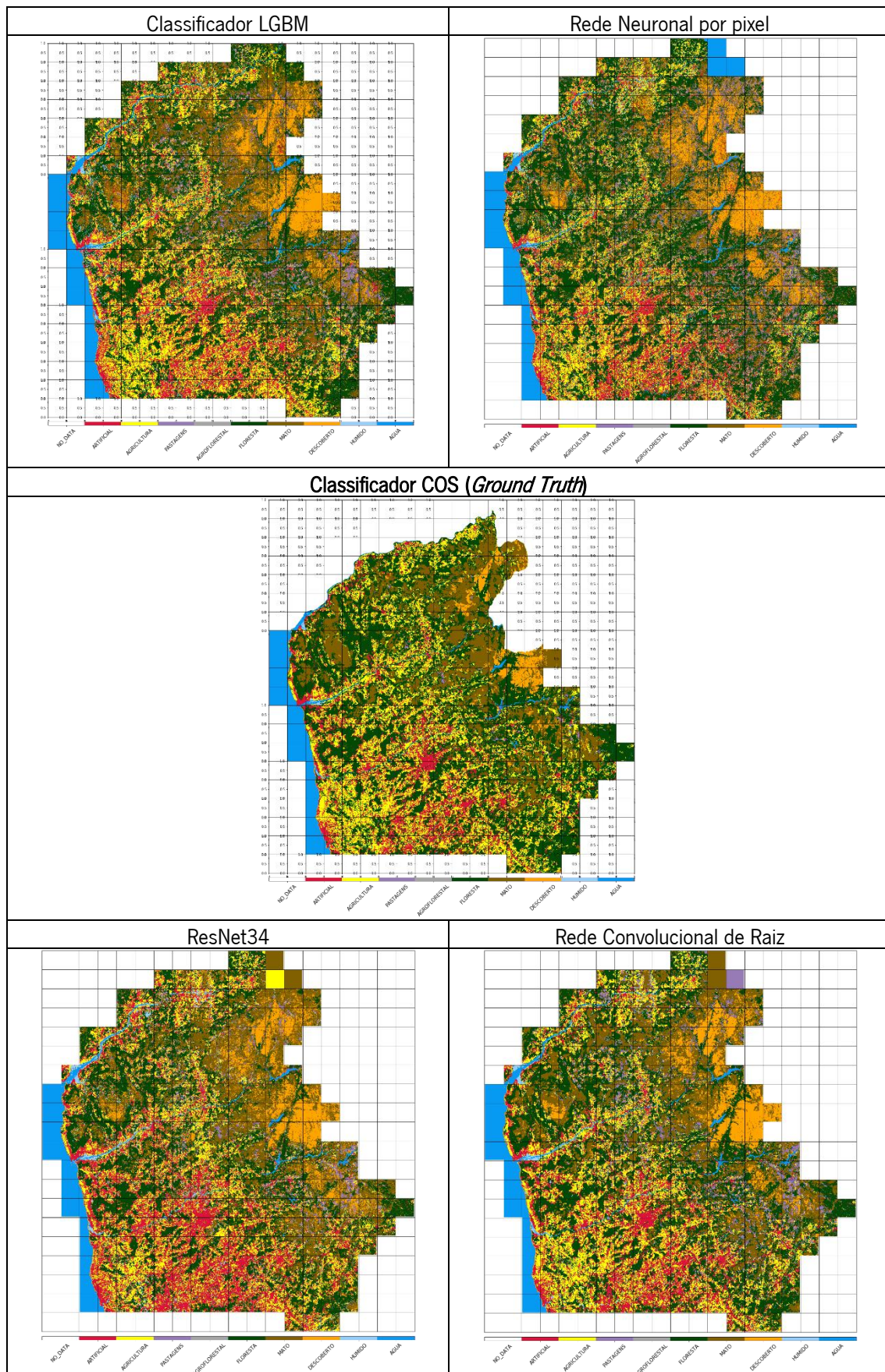


Figura 29 - Comparação da classificação do solo real com a classificação prevista pela rede convolucional de raiz

5.4. Comparação das previsões



6. Conclusão e trabalhos futuros

Terminada a aplicação de técnicas de aprendizagem automática para classificar o tipo de coberto e de uso do solo Português (Minho) e a análise de todos os resultados obtidos, considera-se que as redes e o classificador tiveram um bom resultado, mas que ainda poderia vir a ser melhorado, com mais tempo, recursos e experimentações.

De forma geral, houve pior precisão na classificação das classes agricultura, floresta e mato, uma vez que são áreas muito semelhantes nas imagens de satélite vistas de cima. Chegou-se à conclusão, após a análise de todos os resultados, métricas e matrizes de confusão, que a técnica de aprendizagem que obteve maior sucesso foi a rede convolucional criada de raiz.

As maiores dificuldades na realização deste projeto foi ao criar o *dataset* para as redes neuronais convolucionais, mais precisamente, selecionar os pixels vizinhos para se criar uma imagem. Os recursos de hardware e o tempo necessário para transferir e treinar os dados também se tornou um obstáculo, uma vez que diminuiu a quantidade de testes e de redes que se gostaria de ter experimentado.

Apesar de todas as dificuldades, conseguiu-se chegar a alguns resultados dos vários modelos que se fez. O modelo com melhores resultados foi então a Rede de raiz convolucional, com uma *accuracy* de 80.1%, a seguir o classificador LGBM com 72.5%, a rede Resnet34 com 70.3%, a rede neuronal ResNet50 com uma *accuracy* de 66.2% e por fim a rede neuronal por pixel com 65.1%.

Futuramente poder-se-ia com mais recursos, tentar melhorar o projeto, se fazer para o país inteiro ou até mesmo para outros países. Outro projeto futuro interessante seria escolher vários momentos históricos diferentes de imagens de satélite, para se observar e analisar uma evolução dos cobertos e uso de solos em Portugal.

7. Referencias

1. <https://www.kaggle.com/keras/resnet50>
2. <https://www.tandfonline.com/doi/full/10.1080/15481603.2019.1650447>
3. <https://eo-learn.readthedocs.io/en/latest/>
4. <https://sentinelhub-py.readthedocs.io/en/stable/>
5. <https://github.com/sentinel-hub/eo-learn/>
6. <https://medium.com/sentinel-hub/land-cover-classification-with-eo-learn-part-1-2471e8098195>
7. http://www.dgterritorio.pt/dados_abertos/cos/