Análise ao *dataset – Skin MNIST*

Depois de concluída a análise do *dataset* anterior, *Breast Histopathology*, seguiu-se a análise de outro problema, relacionando o âmbito do trabalho.

O *Skin Cancer MNIST: HAM10000*, é representado por um elevado conjunto de imagens dermatológicas, espelhando diferentes lesões associadas ao cancro da pele[1].

Estas imagens foram obtidas recorrendo a um processo denominado por dermatoscopia. Os dermatologistas recorrem ao dermatoscópio, de modo a conseguir avaliar, com maior rigor e segurança, as lesões do seu paciente. Permitindo-lhe assim identificar a gravidade da lesão sofrida.

Os dermatoscópios permitem aos profissionais de saúde, a visualização de lesões na pele, em escalas muito amplificados, possibilitando a análise de uma lesão, com um aumento de/até 400 vezes, o seu tamanho.

Para além disso, estes aparelhos permitem a digitalização das lesões, para um computador. Ajudando assim, a estabelecer um mecanismo de análise mais segura e gradual (verificação do avanço ou recuo das lesões).

As imagens relativas a este estudo, foram obtidas ao longo de um extenso período de tempo, nomeadamente 20 anos. Contando com o apoio de: o *Department of Dermatology at the Medical University of Vienna*, e ainda com o apoio do Médico/Professor *Cliff Rosendahl*.

O *Skin Cancer MNIST: HAM10000*, apesar de ser um *dataset* recente: disponibilizado em 2018, é já considerado o *standard,* no que toca à análise de lesões associadas ao Cancro da Pele.

São diversas as razões, que ajudam a destacar a sua importância, sendo essencialmente: (1) elevado nº de imagens disponíveis: 10015, (2) elevado nº de classes consideradas: 7 e (3) percentagens de amostras para as classes menos balanceadas, ajustada em conformidade com as possibilidades[1].

São 7 os tipos de lesões identificadas, ao longo do *dataset*. Sendo estas: *akiec*, *bcc*, *bkl, df*, *nv*, *mel* e *vasc*. Estas "*labels"* são apenas identificadores do tipo de lesão associada, porque cada classe pode conter várias patologias associadas à lesão associada[1]. Seguidamente, num contexto resumido, é contextualizada cada uma das classes descritas atrás.

A classe ***Akiec,*** agrega dois tipos de cancro não invasivo: *Actinic Keratoses* e ainda Intraepithelial Carcinoma*. Actinic Keratoses* também conhecidos por *Solar Keratoses*, descrevem lesões causadas pela excessiva exposição solar, sendo que a probabilidade de evolução para cancro invasivo é baixa. *Intraepithelial Carcinoma*, também conhecido por *Bowen's disease*, representa uma lesão semelhante à *Actinic Keratoses*, com a exceção da causa associada, isto é, a lesão é causada por um vírus e não pela exposição solar. Ambas as patologias foram inseridas pelos autores na mesma classe, devido ao facto de ambas serem consideradas e citando [1]: "variantes de squamous cell carcinoma".

BCC é a abreviação de *Basal cell carcinoma,* e representa o tipo mais comum de cancro de pele. As principais causas associadas a esta lesão, são sobretudo: a exposição solar excessiva, pessoas com pele mais branca tem maior risco ou exposições a radioterapias. Normalmente, e quando detetada a tempo, não resulta em morte, ainda assim é necessário ter em consideração, que o tumor cresce e pode-se tornar invasivo.

A classe ***BKL***, *Benign Keratosis* é segundo os autores uma classe genérica, sendo a mesma composta por três sub-grupos, sendo estes: *seborrheic keratoses*, *solar lentigo* e ainda *lichen-planus like keratoses*. Estes três sub-grupos formaram uma única classe, porque e citando os autores: "apresentam semelhanças biológicas e são normalmente avaliadas considerando o mesmo termo histopatológico". Uma curiosidade destas lesões, é que o seu aspeto varia de acordo com o local da lesão[1].

DF, *Dermatofibroma*, representam lesões benignas que podem aparecer na pele, sendo desconhecidas as causas associadas ao seu aparecimento. Na maioria das situações, não existe a necessidade de recorrer a qualquer tipo de cirugia, quer para o seu tratamento, quer para a sua eliminação.

NV, *Melanocytic nevi*, representam neoplasmos benignos, e que podem resultar de uma forma adquirida, como por exemplo: exposição à luz solar, ou ainda de uma forma congénita, isto é, genética. Estas lesões aparecem normalmente, nos primeiros 20 anos de vida dos pacientes. Na maioria dos casos, a probabilidade de evoluir para Melanoma, é reduzida.

Mel, *Melanoma*, representam uma lesão na pele maligna, e que se desenvolve a partir de *melanocytes*. Existem vários tipos de melanomas, como: Superficial, Nodular, Lentigo maligna, ou Acral lentiginoso. Existem várias as causas associadas a esta lesão, mas a que revela uma maior incidência é a excessiva exposição solar. Os índices de cura são muito elevados, quando não existem metástases associadas.

Finalmente, ***Vasc,*** *Vascular skin lesions*, representam lesões que originam problemas nos vasos linfáticos, podendo incluir tumores benignos ou malignos ou malformações. Algumas das variantes consideradas nestes *dataset*: *cherry angiomas*, *angiokeratomas, pyogenic granulomas* e *Hemorrhage*. Algumas destas lesões são facilmente detetadas, contudo outras necessitam de um maior cuidado, na sua análise.

Seguidamente, irá ser demonstrada a análise exploratória, que fora realizada ao *dataset*, com o objetivo de ajudar o autor, a perceber as "peças chave" do problema. Para tal, foi criado um *Jupyter notebook*, para auxiliar na análise e gráficos, que surgiram subsequentemente da análise.

Análise Exploratória

Antes de avançar para a análise dos dados, é necessário explicar um conjunto de passos iniciais, que foram estabelecidos, e que são vitais, durante todo o processo de análise dos dados. Sendo estes: Armazenamento das imagens e Leitura/Adaptação do *Dataframe,* que agrega a informação relativa a cada uma das amostras do problema.

Sendo assim, primeiramente foram transferidos todos os ficheiros associados ao problema em questão, os mesmos encontram-se disponibilizados, na comunidade *Kaggle* [2].

As imagens encontram-se divididas, ao longo de duas pastas. As imagens estão em formato *.jpg* e apresentam um tamanho de 600 pixéis por 450 pixéis, respetivamente largura e altura.

Para além das imagens, foram disponibilizados ainda cinco ficheiros *.csv*. O primeiro ficheiro *HAM10000_metadata*, contêm um conjunto de informações relevantes sobre cada uma das amostras do problema, tais como: idade do paciente, local da lesão, sexo do paciente ou classe associada. Um maior destaque a este documento, irá ser demonstrada através da análise exploratória realizada. Já os restantes ficheiros, descrevem para cada amostra os valores

referentes aos pixéis em cada posição da imagem, considerando um tamanho pré-estabelecido, sendo considerado um tamanho de 8*8 e 28*8. A representação dos pixéis, foi demonstrada num formato *RGB* e ainda *greyscale*.

Os ficheiros e imagens foram devidamente incluídos na raíz do projeto, de modo a facilitar, o processo de leitura dos dados. De salientar ainda, que ao longo do projeto e também da análise exploratória, não foi considerada a utilização de *path's* relativos, de modo a aumentar a flexibilidade do código. Esta preocupação revela-se essencial, porque o código necessita de ser trabalho em diferentes máquinas e diferentes sistemas operativos, evitando assim a necessidade de "alterar configurações de caminhos".

A Figura 1 ilustra a estrutura que agrega os vários ficheiros alusivos ao *dataset* em estudo, e que foram descritos atrás. A Figura 2 enumera várias "variáveis de configuração" estabelecidas no roteamento e definição dos caminhos inerentes aos dados e ficheiros, a analisar.

```
ROOT_DIR = os.getcwd()
print(ROOT_DIR)
INPUT_DIR = os.path.join(ROOT_DIR, config.INPUT_FOLDER)
print(INPUT_DIR)
PATIENTS_INFO = os.path.join(INPUT_DIR, config.INFO_PATIENTS)
print(PATIENTS_INFO)
```

```
C:\Users\gusta\Desktop\Projetos_Python\skin_mnist
C:\Users\gusta\Desktop\Projetos_Python\skin_mnist\input
C:\Users\gusta\Desktop\Projetos_Python\skin_mnist\input\HAM10000_metadata.csv
```

Figura 1 - Estrutura que agrega os vários ficheiros do dataset

```
INPUT_FOLDER = 'input'
IMAGES_ACESS = 'images/*.jpg'
INFO_PATIENTS = 'HAM10000_metadata.csv'
PIXEL_28_RGB_CSV = 'HMNIST_28_28_RGB.csv'
```

Figura 2 - Variáveis de configuração (route)

A raíz do projeto é designada pela pasta *skin_mnist*, observável através da Figura 1, sendo este o ponto "mais distante", da estrutura em árvore do projeto. No interior do projeto, para além dos *scripts* de código, e de outros diretórios, existe uma pasta *input*. É esta pasta apelidada de *Input*, que contém as imagens e os ficheiros *.csv* descritos anteriormente. Como, podemos visualizar através da Figura 1, a o ficheiro *HAM10000_metadata.csv*, está no interior da pasta *Input*.

A Figura 2, descreve os nomes dos ficheiros e diretórias, associadas ao problema em análise. Esta abordagem, permite a eventual alteração da estrutura e/ou caminhos destas pastas e ficheiros com facilidade.

Para terminar, esta análise basta apenas identificar o local onde as imagens são armazenadas. A Figura 3, ilustra o caminho único onde são armazenadas todas as imagens alusivas ao *dataset*.

```
IMAGES_REGEX = os.path.join(INPUT_DIR, config.IMAGES_ACESS)
images_paths = config_func.getImages(IMAGES_REGEX)
images_paths[0]
```

```
'C:\\Users\\gusta\\Desktop\\Projetos_Python\\skin_mnist\\input\\images\\ISIC_0024306.jpg'
```

Figura 3 - Exemplo de um caminho, associado às várias imagens do dataset

Tal como tinha sido referido anteriormente, a pasta *Input* armazena os ficheiros *.csv*, mas também as amostras do problema. Recorrendo, à visualização da Figura 3 é possível verificar que a pasta *Input*, agrega no seu interior uma outra pasta: *images*. Esta pasta, é responsável por armazenar unicamente, todas as imagens do problema. Ou seja, a utilização da pasta *images*, permite isolar as imagens, dos ficheiros *.csv*. Na Figura 3, é ainda possível visualizar o *output* referente a uma imagem: ISIC_0024306*.jpg*. A extensão da imagem, é *.jpg*, tal como já fora exposto anteriormente. Já o indicador ISIC_0024306, representa um identificador associado a cada uma das imagens (este aspecto, volta a ser referido, mais adiante).

Após esta breve contextualização sobre os ficheiros, e também à forma como foram armazenados, segue-se uma análise concreta dos dados.

O primeiro passo aplicado consistiu na leitura do ficheiro *HAM10000_metadata.csv*. Este ficheiro reúne informações específicas, sobre cada imagem presente no *dataset*.

A Figura 4 ilustra as 5 primeiras linhas deste ficheiro.

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear

Figura 4 - Exemplo do conteúdo do ficheiro - HAM10000_metadata.csv

Este ficheiro foi convertido para um objeto *DataFrame* (biblioteca *Pandas*), de modo a facilitar a leitura, escrita, manipulação e análise dos dados.

O *DataFrame* reúne ao todo sete colunas, sendo estas: *lesion_id, image_id, dx, dx_type, age, sex* e *localization*. A coluna *lesion_id* simboliza o id da lesão associada à cada imagem. Ou seja, podem existir várias imagens associadas à mesma lesão, como podemos constatar, através das 2ª imagens da Figura 4. O campo *image_id* é um identificador único associado a cada imagem. A coluna *dx* representa o diagnóstico associado à lesão do paciente (se existirem várias imagens, as mesmas apresentam o mesmo valor neste campo), isto é, o *target* associado a cada uma das imagens. O campo *dx_type* reflete o tipo de diagnóstico, que foi considerado na avaliação da lesão, existindo 4 possibilidades: *Histopathology* (*histo*), *Reflectance confocal microscopy*, *Consensus of at least three expert dermatologists from a single image* (*consensos*), *Lesion did not change during digital dermatoscopic follow up over two years with at least three images* (*follow-up*). Os dois campos seguintes descrevem o sexo e a idade do paciente. Finalmente, a coluna *localization* indica o local do corpo, onde se localiza a lesão, por exemplo: *ear, scalp, foot, neck*, entre outros.

Mas, antes de proceder à análise dos dados, é necessário adicionar uma nova coluna ao *DataFrame,* isto é, é necessário identificar para cada uma das imagens (via *image_id*), qual a imagem que lhe está associada. Sendo que, a nova coluna irá conter o *path* associado, a essa imagem. Como já fora evidenciado atrás, cada imagem reúne um identificador único, *image_id*, sendo possível através desse identificador, verificar qual a imagem associada a cada *image_id*, visto que o nome da imagem, é descrito através desse mesmo identificador. As Figuras 3 e 4 ilustram a presença desse identificador único associado a cada imagem ISIC_*****.

Dessa forma, foi adicionada uma nova coluna no *DataFrame,* "apelidada" de *path*. O resultado final do *DataFrame* está representado através da Figura 5.

	lesion_id	image_id	dx	dx_type	age	sex	localization	path
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp	C:\Users\gusta\Desktop\Projetos_Python\skin_mn...
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp	C:\Users\gusta\Desktop\Projetos_Python\skin_mn...
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp	C:\Users\gusta\Desktop\Projetos_Python\skin_mn...
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp	C:\Users\gusta\Desktop\Projetos_Python\skin_mn...
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear	C:\Users\gusta\Desktop\Projetos_Python\skin_mn...

Figura 5 - DataFrame - após adição de nova coluna: path

Desta forma, o objeto *DataFrame,* já se encontra devidamente formulado, para que se possa dar seguimento à realização de outras tarefas, nomeadamente: análise exploratória (e que vai ser abordada de seguida), e ainda na preparação dos dados do problema (este objeto *DataFrame*, é um "ponto de partida", para a obtenção dos vários conjuntos de dados a utilizar na criação, treino e previsão dos modelos.

Inicialmente, foi efetuada uma breve inspeção ao *DataFrame*, como medida de precaução, e de modo a verificar se tudo estava em conformidade. Sendo assim, existiu a preocupação de: (1) verificar se o nº total de imagens coincidia com o valor indicado em [1], isto é, 10015 imagens no total, (2) confirmação dos atributos do ficheiro *metadata,* com o objeto *DataFrame*, (3) verificação das classes do problema e (4) confirmação de que o *path* de cada imagem está coerente com o seu *id*.

Relativamente ao nº total de imagens presentes no *DataFrame* foi possível concluir que o nº total de linhas no *DataFrame*, é igual ao nº total de imagens declaradas em [1]. Ou seja, o *DataFrame* é composto por 10015 imagens. A Figura 6 enumera o nº total de linhas presentes no objeto.

```
data.shape[0]
```

```
10015
```

Figura 6 - Nº de linhas total do DataFrame

Já, o nº total de colunas também se encontra em conformidade com o ficheiro *metadata*, isto é, o objeto *DataFrame* reúne as 7 colunas presentes no ficheiro: *lesion_id, image_id, dx, dx_type, age, sex, localization*, contando ainda com a nova coluna adicionada *path*. A Figura 7 demonstra as colunas do objeto.

```
data.columns
```

```
Index(['lesion_id', 'image_id', 'dx', 'dx_type', 'age', 'sex', 'localization',
       'path'],
      dtype='object')
```

Figura 7 - Colunas do DataFrame

As classes do problema, tal como expectável estão em conformidade, com o ficheiro *metadata*. Sendo ao todo 7 as classes do problema. A Figura 8 ilustra as "abreviações", já abordadas, e que referem as classes do problema.

```
classes = data.dx.unique()
classes
```

```
array(['bkl', 'nv', 'df', 'mel', 'vasc', 'bcc', 'akiec'], dtype=object)
```

Figura 8 - Labels relativas às classes do problema

Finalmente, foi necessário verificar se o conteúdo inerente à nova coluna adicionada: *path*, estava em concordância com o *image_id*. Isto é, é necessário garantir, que o caminho de cada imagem é o correto, referenciando assim o *image_id*, correto da imagem. A Figura 9 demonstra a relação entre ambas as colunas, *image_id* e *path*.

	image_id	path
0	ISIC_0027419	C:\Users\gusta\Desktop\Projetos_Python\skin_mnist\input\images\ISIC_0027419.jpg
1	ISIC_0025030	C:\Users\gusta\Desktop\Projetos_Python\skin_mnist\input\images\ISIC_0025030.jpg
2	ISIC_0026769	C:\Users\gusta\Desktop\Projetos_Python\skin_mnist\input\images\ISIC_0026769.jpg
3	ISIC_0025661	C:\Users\gusta\Desktop\Projetos_Python\skin_mnist\input\images\ISIC_0025661.jpg
4	ISIC_0031633	C:\Users\gusta\Desktop\Projetos_Python\skin_mnist\input\images\ISIC_0031633.jpg

Figura 9 - Relação entre as colunas Image_ID e Path

Como podemos observar através da Figura 9, o identificador presente no *path* das imagens (*ISIC_****.jpg*) é coerente com o seu *id*, garantindo assim a concordância, entre o *id* de uma imagem, e o seu caminho.

Posto isto, segue-se a análise concreta aos dados do problema. Esta análise irá ser apoiada com recurso à disponibilização de vários gráficos, que ajudam a entender melhor os dados, e a identificar padrões/particularidades nos dados.

Primeiramente, foi efetuada uma procura por valores em falta no *dataset*. A Figura 10 demonstra, um relatório genérico sobre o *dataset*.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10015 entries, 0 to 10014
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   lesion_id     10015 non-null  object
 1   image_id      10015 non-null  object
 2   dx            10015 non-null  object
 3   dx_type       10015 non-null  object
 4   age           9958 non-null   float64
 5   sex           10015 non-null  object
 6   localization  10015 non-null  object
dtypes: float64(1), object(6)
memory usage: 547.8+ KB
```

Figura 10 - Análise de algumas propriedades do dataset

Observando a Figura 10, é possível concluir que todas as colunas, à exceção da idade, não apresentam valores em falta (o que não quer dizer, que não hajam incoerências nos dados, como vamos verificar adiante). Relativamente, ao atributo *age*, é possível constatar a presença de alguns valores em falta, mais concretamente 57 imagens, não têm associada uma idade.

Para o problema em causa, não existe a necessidade de imputar valores nesta coluna, dado que a inexistência das idades dos pacientes não invalida a utilização das imagens, visto que o objetivo

do estudo é classificar as imagens, sendo que a idade não impede a sua análise. Da mesma forma, é também descurada a eventual eliminação destas 57 amostras, visto que iria existir uma redução do nº de amostras, e não existe essa necessidade, porque o conteúdo não é afetado, pela falta destes valores.

Ainda, no que toca à análise do campo idade, foram analisadas algumas métricas, tais como: média, valor mínimo e máximo, desvio padrão, etc. A Figura 11 enumera os valores relativos às métricas analisadas.

```
count    9958.000000
mean       51.863828
std        16.968614
min         0.000000
25%        40.000000
50%        50.000000
75%        65.000000
max        85.000000
```

Figura 11 - Análise métricas - campo idade

É possível constatar que a média de idades, está na casa dos 52 anos, algo que está em conformidade com a incidência real, deste tipo de lesões. O desvio padrão é elevado, indicando-nos assim uma elevada disparidade dos dados, o que é visível através dos valores máximos e mínimos (0 anos e 85 anos). Relativamente à relação entre o 2º e o 3º quartil, é possível constatar uma distância semelhante entre a mediana e o 2º e 3º quartil, 10 e 15 respetivamente.

O gráfico ilustrado através da Figura 12, descreve a distribuição das imagens, pelo sexo do paciente.

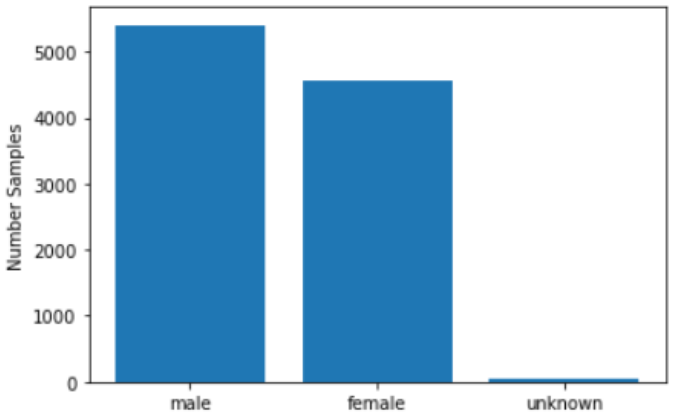

Figura 12 – Distribuição das amostras pelo sexo do paciente

Tal, como já tinha sido destacado anteriormente, não existem apenas incoerências associadas à coluna idade. O atributo sexo, também reúne um pequeno conjunto de amostras, onde a sua categoria, não é valida, sendo esta "unknown". Como podemos visualizar através do gráfico o nº de amostras categorizadas como "unknown" é baixo. O contexto abordado para a coluna idade, mantêm-se válido também para esta coluna. Dado que, esta "indefinição" do atributo sexo em algumas imagens, não interfere com a sua classificação.

Relativamente às categorias "male" e "female", é possível constatar que o nº de imagens associadas a pacientes "homens", revela-se um pouco superior face a pacientes "mulheres". Esta diferença é de cerca de 1000 amostras.

A análise prosseguiu com a exploração, da distribuição do nº de amostras, tendo em conta o local onde a lesão se situa. A Figura 13 ilustra essa mesma distribuição.

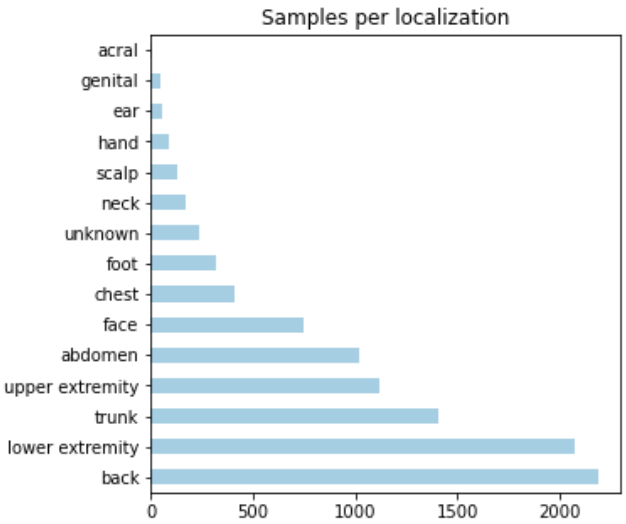

Figura 13 - Nº de imagens, por local onde a lesão se situa

Recorrendo à observação da Figura 12, constatamos a disparidade no nº de amostras, que existe entre as várias localizações das lesões.

Para além disso, constatamos novamente a presença de uma categoria, que não está em coerência com o atributo em análise, mais uma vez: "unknown". Neste caso, o nº de amostras associadas a esta categoria, é significativamente superior, aos valores identificados na coluna idade ou sexo. Existindo assim, 300 a 400 imagens, que não têm associada uma localização válida para a respetiva lesão que identificam. Ainda assim, as conclusões salientadas para as colunas analisadas anteriormente, mantêm-se válidas também para este atributo. Dado que, a inexistência de uma localização válida, não invalida a classificação da imagem.

Analisando as restantes categorias, é possível verificar que existem uma evidente disparidade entre estas, por exemplo: a categoria "acral" reúne um nº extremamente baixo de imagens. Por outro lado, existem mais de 2000 imagens, onde a lesão se situa nas costas "back". Existem 5 classes que juntas agregam quase todas as amostras do *dataset*: *back, lower extremity, trunk, upper extremity* e *abdómen*.

O próximo atributo a analisar é o *dx_type*. Tal como já fora indicado previamente, esta coluna descreve o tipo de diagnóstico, que fora considerado na avaliação de uma determinada amostra. De relembrar ainda, que os tipos de avaliação considerados foram: *Histopathology* (*histo*), *Reflectance confocal microscopy (confocal)*, *Consensus of at least three expert dermatologists from a single image* (*consensos*), *Lesion did not change during digital dermatoscopic follow up over two years with at least three images* (*follow-up*).

A Figura 14 ilustra o nº de amostras, por tipo de diagnóstico.

Da análise da Figura 14 concluímos, que existem dois diagnósticos com bem maior incidência, perante as amostras do problema, concretamente: *Histopathology* e *follow-up*. Ou seja, na maioria das situações, não foi necessário recorrer a outras vias de diagnóstico, para aferir qual o tipo de lesão de um paciente. Ainda assim, em cerca de 1000 imagens existiu a necessidade de recorrer à opinião de três ou mais médicos, para aferir com segurança, qual o tipo de lesão do paciente, demonstrando assim a dificuldade, que existiu na classificação "humana" destas

1000 imagens. O recurso ao método *Reflectance confocal microscopy*, apenas se revelou necessário num baixo nº de casos.

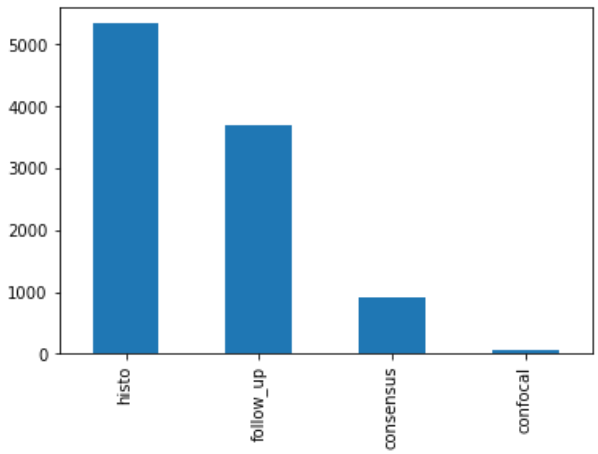

Figura 14 - Nº de amostras, por tipo de diagnóstico

Finalmente, segue-se a análise da distribuição das imagens, pelas várias classes do problema. A Figura 15 ilustra essa variação.

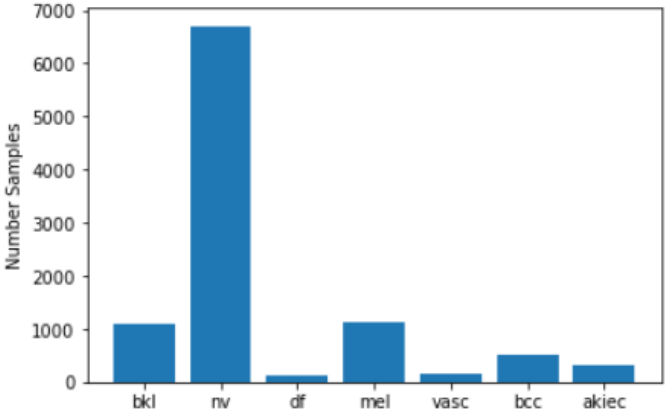

Figura 15 - Nº de imagens, por classe

Relativamente, à distribuição do nº de imagens por classe, é possível verificar que o *dataset* se revela pouco balanceado, tal como já era expectável. O artigo [1] compara a distribuição do nº de imagens por classe deste problema, com outros *dataset's* inseridos no mesmo âmbito. Dessa análise foi possível comprovar, as mesmas evidências: o não balanceamento das classes e ainda o facto das classes: *Melanocytic nevi, Melanoma* e *Benign Keratosis* serem sempre as mais representativas.

Este *dataset*, ao apresentar um maior nº de amostras, dificulta ainda mais o processo de classificação das imagens, dado que a classe *Melanocytic nevi* contêm quase 70% do nº total de imagens do *dataset*. As classes *Melanoma* e *Benign Keratosis,* agregam respetivamente 10% do nº total de imagens, já as restantes quatro classes contêm os últimos 10%.

Desta análise, fica subjacente a eventual dificuldade, que irá ser enfrentada, no processo de classificação das imagens relativas às classes menos balanceadas do problema. Dado que, o seu baixo nº de amostras dificulta a identificação de particularidades próprias destas classes.

Concluída a análise univariada do problema. Seguiu-se uma breve análise bivariada das *features* do problema, tentando apenas relacionar as classes do problema, com algumas das *features*, de modo a entender um pouco melhor o problema (não foi aplicada, uma análise severa, visto que o estudo a realizar não o necessita).

A primeira abordagem considerada, relaciona a idade dos pacientes com a lesão sofrida (classe). A Figura 16 ilustra a variação da idade dos pacientes, por cada uma das sete classes do problema.

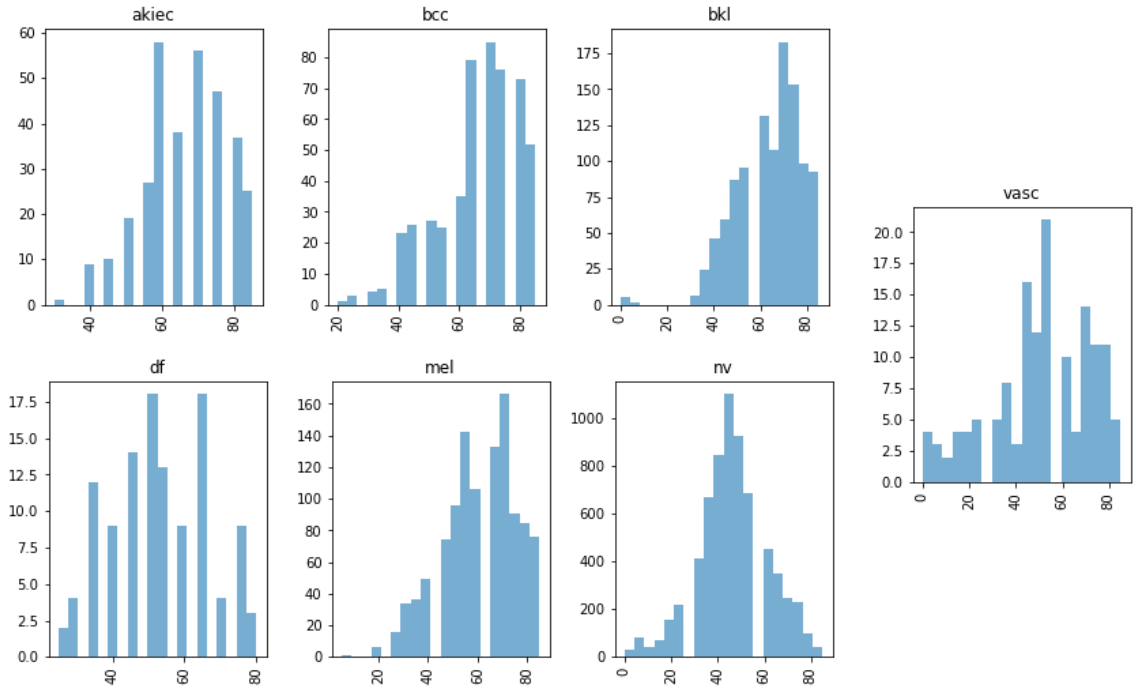

Figura 16 - Distribuição do nº de amostras por idade e por classe

Na maioria das classes é possível observar o mesmo padrão, isto é, existe uma maior incidência das lesões, em pacientes acima dos 50 anos de idade, mais concretamente nas classes: *akiec, bcc, bkl, mel e vasc*. Relativamente, às duas classes restantes: *df* e *nv*, verifica-se um maior equílibrio entre ambas as partes, ou seja, a incidência de casos revela-se semelhante para pacientes com idade inferior e superior a 50 anos. Mas, num contexto geral, independentemente do tipo de lesão sofrida, a probabilidade do paciente ter uma idade superior a 50 anos é elevada.

Seguidamente, foi efetuada uma análise à distribuição do nº de imagens por sexo, para cada uma das sete classes. A Figura 17 ilustra essa distribuição, por cada uma das classes.

Observando a Figura 17, é possível concluir que na maioria das classes, a categoria *male* é a mais representada. A única classe onde, existe uma maior incidência no sexo feminino é a *vasc*. Curiosamente, a classe com maior nº de imagens: *nv*, apresenta uma relação muito próxima, entre ambos os sexos. Demonstrando assim, que as mulheres também apresentam uma elevada incidência de apresentarem a lesão mais frequente de cancro da pele. As classes *bkl* e *nv* apresentam um nº reduzido de imagens associadas à classe "*unknown*" (este problema já foi devidamente fundamentado anteriormente).

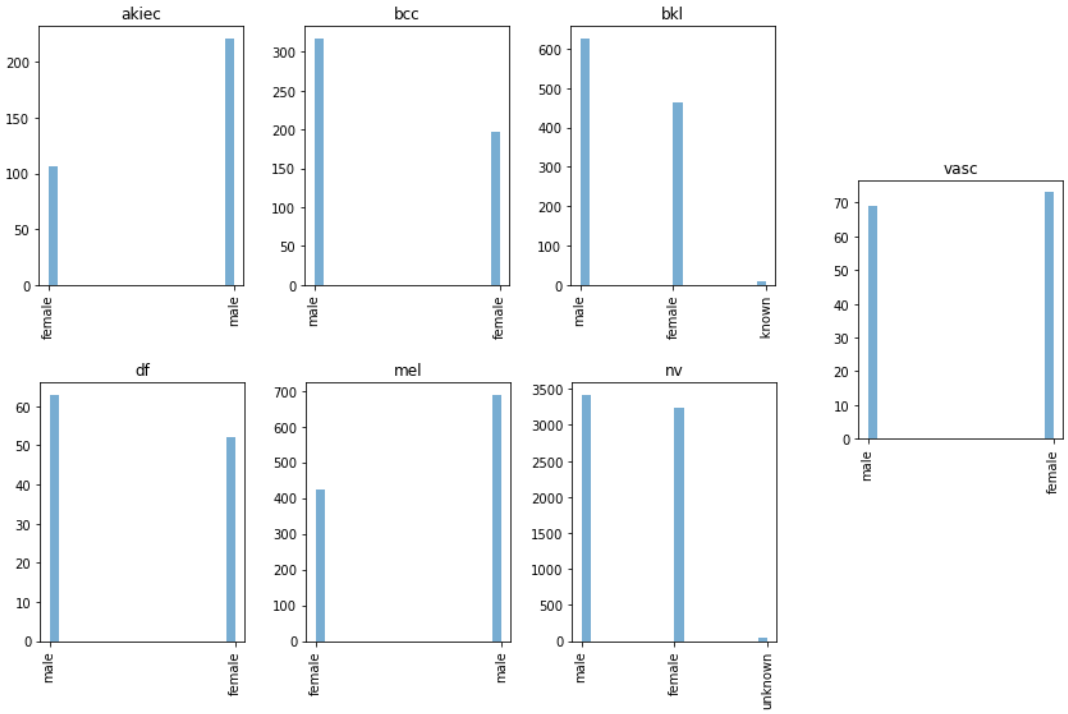

Figura 17 - Distribuição do nº de imagens por sexo, e por classe

A Figura 18, ilustra o nº de imagens, que estão associadas aos vários locais, onde se encontram as lesões, estando agrupadas pelas sete classes do problema.

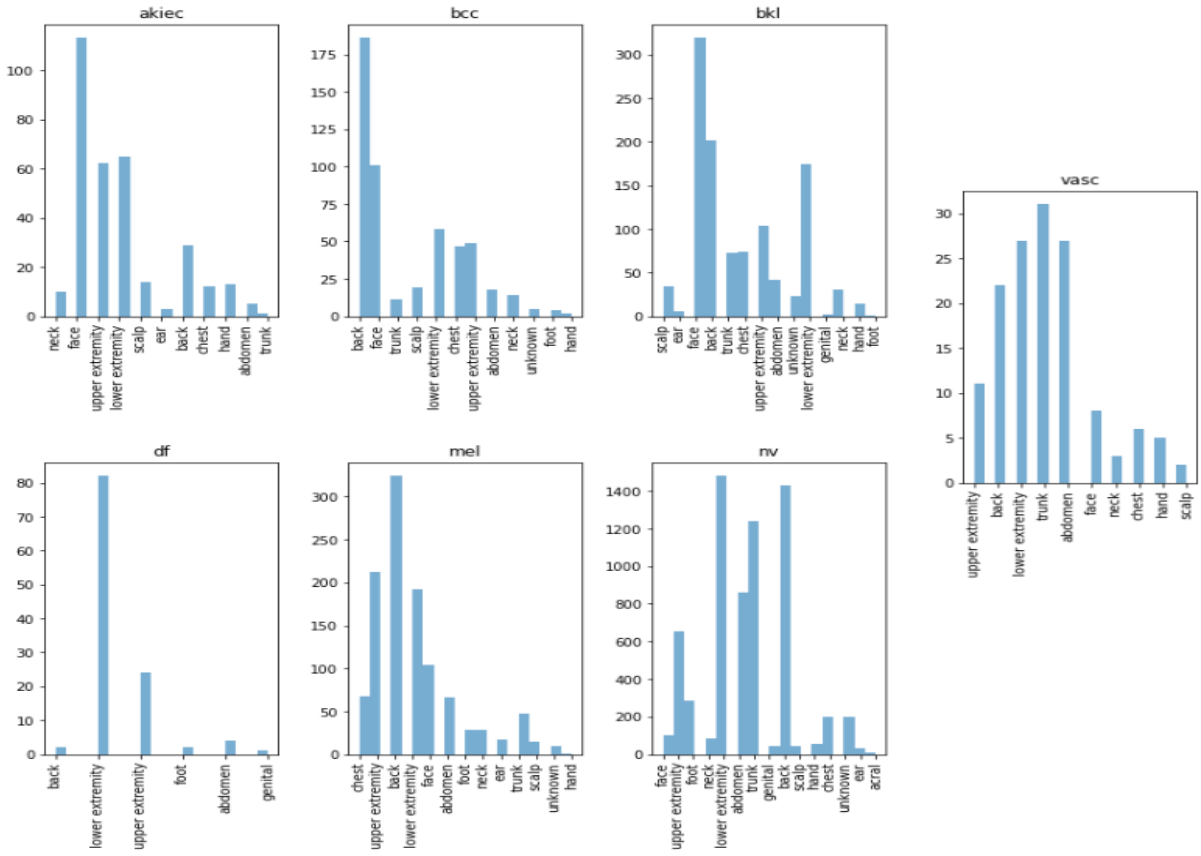

Figura 18 – Distribuição do nº de classes por localização, e por classe

Analisando, a Figura 18 é possível comprovar que os locais com maior predominância de lesões, estão dependentes do tipo de lesão sofrida pelo paciente. Isto é, observando por exemplo a categoria *back,* constatamos que a mesma é muito saliente na maioria das classes, como: *bcc, bkl, nv, mel* ou *vasc*. Contudo, nas classes *df* ou *akiec* não se revela muito dominante face às restantes categorias.

É possível ainda concluir que na maioria das classes, existem locais que estabelecem uma "área dominante", promovendo assim uma maior incidência de uma lesão se contrair num determinado local. Ou seja, exemplificando com a classe *df,* podemos constatar que as lesões são praticamente apenas inferidas, em 2 locais: *upper* e *lower extremity*. O mesmo se procede com as restantes classes, onde é notória a maior incidência em determinados locais.

Finalmente, o último passo desta análise exploratória incide na ilustração de algumas imagens referentes a cada uma das classes do problema.

Antes de avançar na execução de novas tarefas, é importante visualizar algumas imagens do *dataset*, de modo a tentar identificar algumas particularidades e diferenças, entre as várias classes do problema. Pois dessa forma, torna-se mais fácil dar resposta a uma série de questões, que são pertinentes, tais como: (1) É difícil encontrar diferenças entre as várias classes? (2) Será necessário considerar uma arquitetura mais complexa para a resolução do problema?

Esta análise ajudará futuramente, na definição dos modelos a serem criados, para a classificação de imagens.

Para tal, foi efetuada a leitura do ficheiro *HMNIST_28_28_RGB.csv*. Este ficheiro contém os valores *RGB* de cada imagem, considerando uma altura e largura igual a 28. Para além disso, contém ainda uma coluna, que especifica o *output* associado a cada imagem. **De destacar, que o estudo não vai considerar esta escala, irá considerar uma escala maior** (foi considerada esta abordagem aqui, de modo a reduzir o tempo desperdiçado).

A Figura 19 demonstra o *output* obtido da leitura do ficheiro (*DataFrame* resumido).

	pixel0000	pixel0001	pixel0002	pixel0003	pixel0004	pixel0005	pixel0006	pixel2351	label
0	192	153	193	195	155	192	197	177	2
1	25	14	30	68	48	75	123	27	2
2	192	138	153	200	145	163	201	117	2
3	38	19	30	95	59	72	143	15	2
4	158	113	139	194	144	174	215	92	2

Figura 19 - Leitura do ficheiro HMNIST_28_28_RGB.csv, e respetivo output (DataFrame)

A Figura 19, demonstra assim as cinco primeiras linhas, do objeto *DataFrame,* resultante da leitura do ficheiro indicado. O resultado obtido consiste num objeto que contém 10015 imagens (nº de linhas), e um nº total de colunas igual a 2352. As primeiras 2351 colunas correspondem aos valores dos pixéis *RGB*, expressos numa escala de 0-255. O nº 2351 resulta da multiplicação entre a altura e largura de cada imagem (28*28), sendo que é ainda necessário multiplicar este valor pelo nº total de *channels*, neste caso o nº de *channels* é igual a 3, visto que é utilizado o filtro *RGB* (28*28*3). Finalmente, a última coluna identifica o *output* de cada imagem (o tipo de lesão).

Depois de obtido o *Dataframe,* já estão definidos todos os requisitos necessários, à ilustração de algumas imagens alusivas às várias classes do problema. Sendo assim, a Figura 20 ilustra 3 imagens pertencentes a cada uma das sete classes do problema.

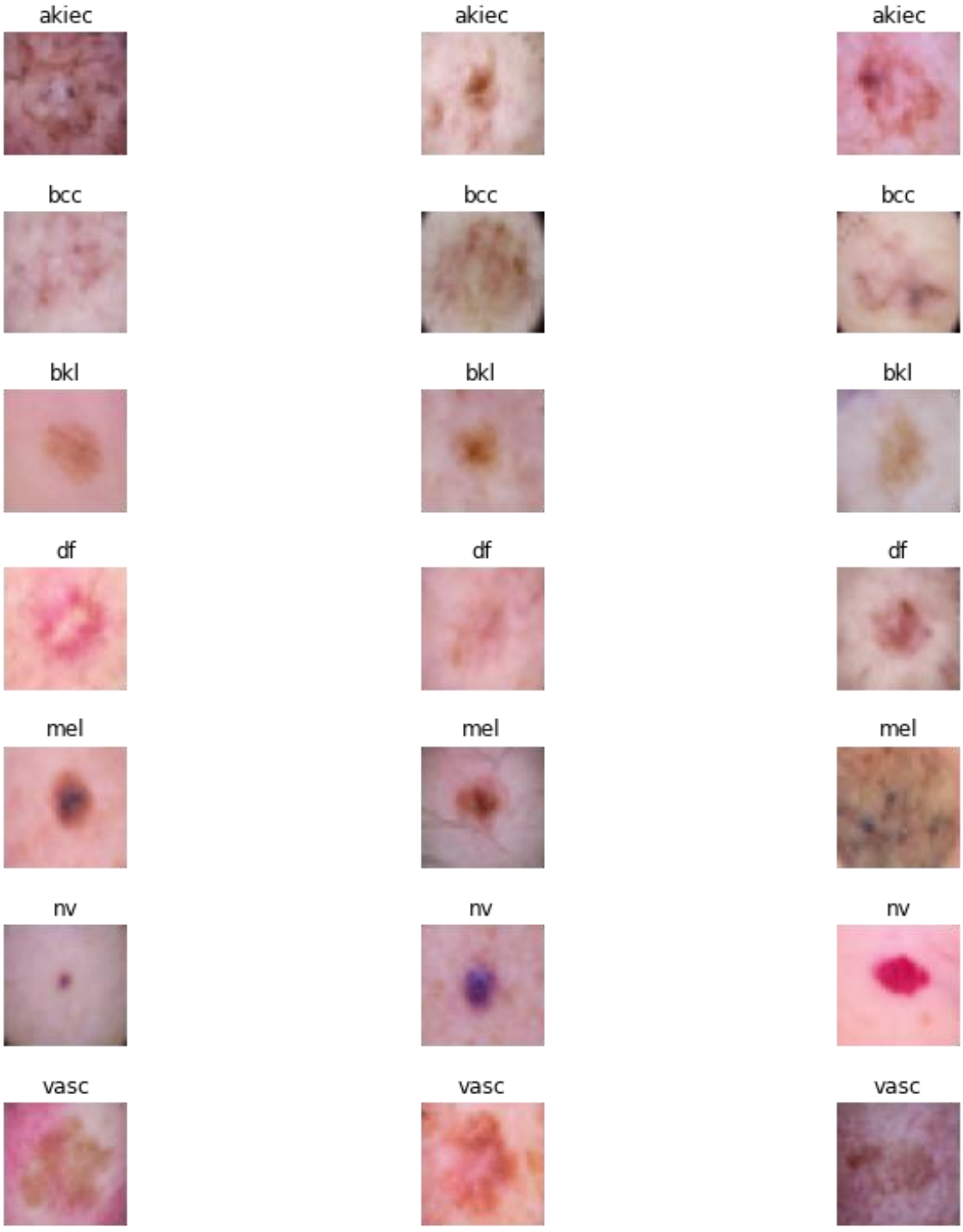

Figura 20 - Imagens alusivas, às várias classes do problema

Tal como já esperado, o problema revela-se complexo. Sendo difícil à 1ª vista identificar particularidades nas imagens, que permitam a fácil e correta classificação das imagens.

Ainda assim, é possível concluir que existem algumas evidências, que ajudam a identificar a classe associada a uma determinada imagem, como: a cor, tamanho ou forma da lesão.

Por exemplo, as lesões associadas à classe *nv* revelam uma cor e forma distintas, face aos restantes tipos de lesões. Isto é, revelam tons fora do "padrão" castanho, e apresentam uma lesão em forma circular, onde o seu tamanho é pequeno.

Já nas classes *vasc* ou *akiec*, a forma da lesão revela-se menos uniforme, não seguindo o padrão circular da classe *nv*. A cor da lesão já segue o padrão visível ao longo de todas as classes, isto é, lesão com tons castanhos.

Já as restantes classes: *mel, bcc, bkl* e *df* apresentam bastantes semelhanças entre si, sendo difícil identificar particularidades únicas, em cada uma das classes mencionadas. Mas, ainda assim é percetível, por exemplo que a classe *bkl* apresenta tons mais acastanhados do que a classe *df*, e as suas lesões apresentam tamanhos inferiores, à classe *bcc*. Ou seja, apesar de ser bastante complexa a classificação correta destas imagens (daí a dificuldade presenciada pelos profissionais de saúde, sendo necessário recorrer a métodos, como: o *follow-up*, onde é necessária a opinião de diversos médicos no diagnóstico de uma lesão), existem alguns factores que ajudam a identificar qual a classe correta.

Técnicas de Pré-Processamento aplicadas:

Concluído o estudo do *dataset*, segue-se a realização de um determinado conjunto de tarefas de processamento de dados. Estas tarefas, resumem-se essencialmente à aplicação de técnicas de obtenção, divisão, limpeza e transformação dos dados.

Esta fase deve ser analisada e definida com ponderação, pois os resultados obtidos no futuro, estarão diretamente relacionados, com as práticas implementadas previamente. Sendo assim, e tendo em conta o *dataset* em estudo, é necessário dar maior enfâse às técnicas de obtenção, divisão e transformação dos dados. As técnicas de Limpeza não se aplicam a este contexto, dado que o autor irá trabalhar diretamente com as imagens, e não com as informações (*metadata*), que foram exploradas durante a análise exploratória.

Tal como já fora referido atrás, o tamanho real das imagens é de 600*450, respetivamente largura e altura. Mas, os ficheiros *.csv* fornecidos consideram apenas, a utilização de formatos muito pequenos (28*28 e 8*8), reduzindo assim a qualidade das imagens, dado que muita informação foi comprimida/extraída. Sendo assim, foi necessário proceder à aplicação de técnicas de *resize,* das imagens reais. Os valores para a altura e largura consideradas (das imagens a redimensionar), foram de: 128 e 128 respetivamente.

Neste processo foi utilizada a biblioteca *opencv*, que permite a obtenção dos valores dos pixéis das imagens, tendo em conta o formato da imagem pretendida. De realçar ainda que foi mantido o mesmo filtro (das imagens originais), isto é, o filtro *RGB*.

Desta operação resultaram dois *arrays:* o 1º contendo os valores *RGB*, dos pixéis de cada imagem (no formato redimensionado: 128*128), e ainda um 2º *array* contendo os *targets* associados a cada imagem.

Estes dois *arrays*, agregam todos os dados do problema. Dessa forma, é necessário efetuar uma divisão das amostras totais, pelos vários conjuntos de dados, a serem utilizados posteriormente, em tarefas de treino e previsão dos modelos, ou seja, dividir o nº total de imagens disponíveis em três conjuntos de dados: treino, validação e teste.

A divisão aplicada representa o *standard*, isto é, 60%-20%-20% respetivamente treino, validação e teste. Por exemplo, considerando um nº total de imagens igual a 2000, a distribuição dos dados será de: 1200, 400, 400, respetivamente conjunto de treino, validação e teste.

Após a aplicação das técnicas de obtenção e divisão dos conjuntos de dados necessários, segue-se a aplicação de técnicas de transformação dos dados.

Este tipo de técnicas é muito importante, neste tipo de problemas, visto que a aprendizagem dos modelos está dependente dos dados. Ou seja, é necessário garantir que existe uma escala comum entre as várias *features* do problema, neste caso entre os vários pixéis. Pois, a escala destes valores varia entre 0-255, isto é, pixéis com valores mais elevados revelam uma maior "importância", comparativamente a pixéis que apresentem valores mais baixos.

Dessa forma recorreu-se à aplicação da técnica *z-score*, mais conhecida por standardização, em que a mesma tenta atribuir uma escala comum entre as várias *features* do problema. Para tal, a mesma aplica a diferença entre o valor de uma feature *X*, e o valor médio presente no conjunto de dados a considerar. O resultado desta subtração, é depois dividido pelo desvio padrão do conjunto de dados. De uma forma resumida, esta técnica redimensiona a transformação de um determinado conjunto de valores, para que a média seja 0 e o desvio padrão igual a 1. A Fórmula 1 ilustra a equação descrita.

$$X_{new} = \frac{X - \mu}{\sigma} \quad (1)$$

Esta técnica foi aplicada a todos os conjuntos de dados: treino, validação e teste. Mas, e de modo a evitar *data leakage*, antes de aplicar a Fórmula 1, é necessário calcular a média e desvio padrão do *dataset* de treino. Ou seja, de modo a evitar que os *dataset's* de validação e de teste, reúnam informações adicionais que não são desejadas, é necessário garantir que por exemplo: o *dataset* de teste mantêm unicamente informação "real", e não informação "conhecida previamente".

Sendo assim, a Fórmula 1 aplicada no conjunto de dados de validação e de teste, recorre à utilização da média e desvio padrão, dos dados de treino. Desta forma, garante-se que os modelos generalizam apenas dados que nunca antes foram vistos.

Finalmente, a última operação a aplicar consiste na transformação do *output* das imagens¸ em formato binário (*one-hot encoding*).

Análise de Resultados:

O próximo "passo" a desenvolver, baseia-se na criação de modelos convolucionais. Estes modelos irão ser utilizados, como "ferramenta" de aprendizagem dos dados que foram anteriormente "tratados".

Tal, como aconteceu no estudo do *dataset* anterior (*breast histopathology*), irão ser aplicados vários tipos de redes convolucionais, com o objetivo de comparar e aferir os benefícios, que estas promovem.

De modo a melhor a eficiência dos modelos, irá ser considerada a aplicação de várias estratégias de treino, sendo expectável que as mesmas promovam melhorias na aprendizagem e generalização dos modelos.

A análise descrita seguidamente, irá recorrer a diversas ilustrações, com o objetivo de garantir uma análise mais simplista, e visualmente agradável. Por outro lado, a análise de resultados, recorrendo a várias Figuras ajuda a entender quais os pontos a melhorar e quais as melhorias atingidas.

Por outro lado, esta análise tenta ser o mais completa possível, tentando assim promover um estudo gradual do *dataset*, bem como a aplicação de diferentes cenários, considerando os vários modelos aplicados. Ou seja, os cenários a aplicar enquadram-se na aplicação de várias condicionantes: (1) aumento da complexidade da rede, (2) aumento da profundidade da rede, (3) aplicação de estratégias de treino e (4) recursos a algoritmos de otimização, na melhoria dos resultados obtidos.

Como referi anteriormente, esta análise irá ser gradual, isto é, por exemplo: inicialmente é necessário verificar o comportamento dos modelos quando considerado um baixo nº de parâmetros (baixa complexidade), só depois é que o autor deve aumentar a complexidade da rede. Mediante os resultados obtidos, e num contexto faseado, deve-se aplicar os restantes cenários enumerados, na tentativa de melhorar os resultados obtidos, e reduzir os custos computacionais exigidos.

A aplicação de algoritmos de otimização, representa uma abordagem que "concatena" os outros três cenários identificados, tentando dessa forma identificar um modelo eficiente, robusto e com o menor custo computacional, sem a necessidade de trabalho manual, e de tentativas incrementais, de melhoria de resultado.

Finalmente, importa apenas realçar que as arquiteturas *CNN* consideradas foram: *AlexNet,* VGGNet e *U-Net*. Já, as estratégias de treino utilizadas foram: *Oversampling, Data Augmentation* e *Segmentation Mask's*. Já as métricas de avaliação consideradas foram: *recall, precision, accuracy* e *f1-score* (quer numa análise geral – ponderação multi-classe, e ainda análise única das classes).

Seguidamente, é efetuada a análise dos resultados obtidos (sendo que, a análise de cada rede *CNN*, é separada).

AlexNet:

O primeiro tipo de rede em análise, é a rede *AlexNet*. Posteriormente, é efetuado um estudo das várias condicionantes descritas atrás.

Importa apenas salientar, que algumas das representações contextualizadas seguidamente, correspondem a circunstâncias obtidas, após o estudo prévio e iterativo de outras situações referentes ao mesmo contexto. Mas, como não é possível relatar todas as "ocorrências" criadas, então foram apenas documentadas as situações mais relevantes.

Todas as amostras disponíveis foram utilizadas, ao longo dos vários cenários aplicados, visto que este valor já é baixo (10015 imagens), não fazendo assim sentido limitar ainda mais este valor.

Análise à complexidade da rede:

O primeiro cenário a considerar prende-se com a aplicação de diversos cenários, relacionados com a variação da complexidade da arquitetura, mais concretamente o aumento ou a diminuição do nº total de parâmetros da rede.

É essencial efetuar este estudo, numa fase precoce da análise, visto que permite verificar o comportamento do modelo, quando se considera uma rede mais complexa. Esta análise, permite aferir qual a abordagem que permite a obtenção dos melhores resultados. Sendo que, a aplicação das estratégias de treino, estão dependentes da arquitetura considerada (complexidade, profundidade, *etc*).

Rede de Baixa Complexidade:

Sendo assim, o primeiro exemplo aplicado considera a aplicação de uma rede com baixa complexidade, isto é, um nº baixo de parâmetros de treino.

O modelo é constituído por três redes convolucionais e ainda duas *Dense layers*, sendo que a segunda representa a camada de *outputs*. De modo, a garantir que a rede apresenta um "baixo custo", recorreu-se à utilização de um baixo nº de filtros e de neurónios. No final, o modelo reuniu 91000 parâmetros de treino, um valor baixo, considerando o tamanho das imagens (128*128).

Tal como já referido anteriormente, foram utilizadas todas as imagens disponibilizadas. Sendo que, para já não se recorreu à utilização de qualquer estratégia de treino, na tentativa de otimizar os resultados obtidos (mais tarde, essas estratégias irão ser consideradas).

Da aplicação deste exemplo, não se espera a obtenção de resultados muito satisfatórios, dado que o problema mantêm-se não "balanceado", poucas amostras consideradas (necessário considerar estratégias de aumento deste nº) e muito provavelmente para resolver este problema é necessário recorrer a redes mais complexas e mais profundas, dada a sua complexidade.

As Figuras 21 e 22 ilustram respetivamente os valores referentes às métricas consideradas e a matriz confusão relativa a todas as classes. Já as Figuras 23 e 24 descrevem a variação do custo e da *accuracy*, ao longo das *epochs* respetivamente.

	precision	recall	f1-score	support
akiec	0.34	0.39	0.36	64
bcc	0.56	0.24	0.33	105
bkl	0.50	0.31	0.39	213
df	0.43	0.16	0.23	19
mel	0.52	0.33	0.40	210
nv	0.81	0.96	0.88	1366
vasc	1.00	0.27	0.42	26
accuracy			0.75	2003
macro avg	0.59	0.38	0.43	2003
weighted avg	0.72	0.75	0.72	2003

Figura 21 – Resultados das métricas

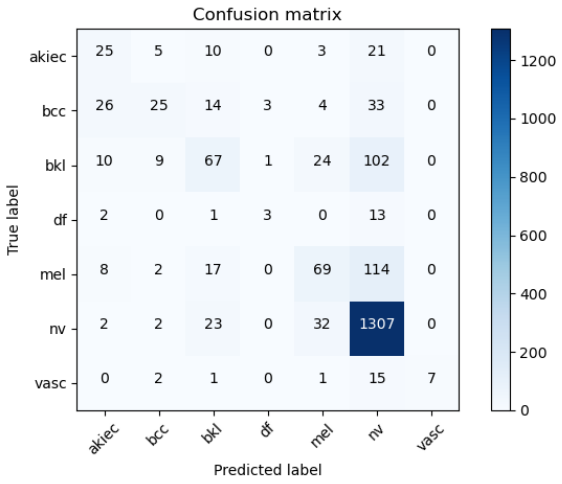

Figura 22 – Matriz confusão

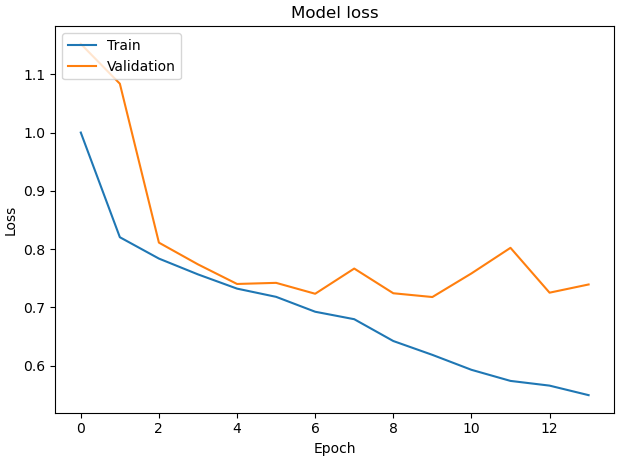

Figura 23 – Variação do custo, ao longo das epochs

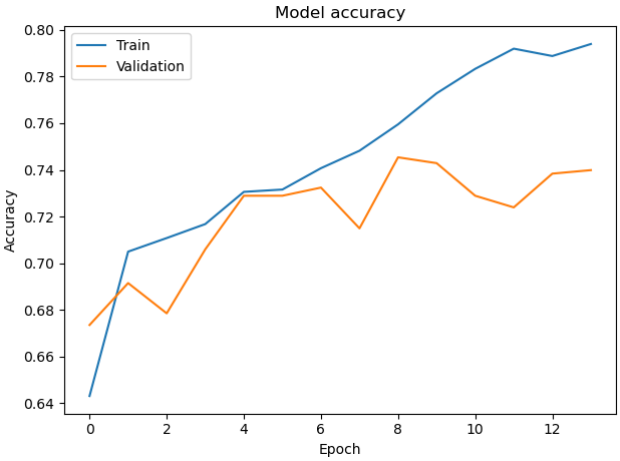

Figura 24 – Variação da accuracy, ao longo das epochs

Analisando as Figuras 21 e 22, é possível comprovar que os resultados obtidos, tal como expectável, estão longe do pretendido. O valor final da *accuracy*: 75%, é muito enganador, porque observando a Figura 22, é possível concluir que o modelo apenas classificou

corretamente as imagens da classe mais representada do problema, *Melanocytic nevi* (*nv*). Já, nas restantes classes, o modelo revelou muitas dificuldades de aprendizagem.

Da análise, às Figuras 23 e 24 é possível constatar a dificuldade de generalização do modelo, que está diretamente relacionada com: a disposição atual dos dados (amostras não balanceadas, poucas amostras), e à arquitetura atual do modelo, isto é, a sua baixa complexidade e profundidade, dificultam o processo de aprendizagem do modelo.

Rede de Elevada Complexidade:

Da mesma forma, foi considerada a aplicação de um modelo mais complexo, ou seja, a arquitetura definida no exemplo anterior manteve-se, mas o nº de parâmetros da rede aumentou significativamente.

É importante comparar os resultados obtidos neste exemplo, com o exemplo anterior, de modo a conseguir perceber, qual a complexidade do modelo que é exigida, de modo a extrair o máximo de conhecimento dos dados.

Isto é, mediante a complexidade do problema e do nº de dados do mesmo, identificar uma rede que garanta a aprendizagem e generalização adequada dos dados. Contudo, e antes de proceder à análise de outros fatores, é importante identificar qual o nº de parâmetros da rede, que são necessários, de modo a garantir um modelo robusto e capaz, de aprender corretamente os dados.

Sendo assim, foi considerada a mesma arquitetura, mas considerando um maior nº de filtros e de neurónios. É expectável, uma melhoria dos resultados face ao exemplo anterior.

Da mesma forma, as Figuras 25 e 26 ilustram respetivamente as métricas resultantes da avaliação do modelo, e a matriz confusão. Já, as Figuras 27 e 28 representam a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.45	0.41	0.43	64
bcc	0.49	0.32	0.39	105
bkl	0.59	0.33	0.42	213
df	0.00	0.00	0.00	19
mel	0.51	0.31	0.39	210
nv	0.81	0.96	0.88	1366
vasc	0.76	0.50	0.60	26
accuracy			0.76	2003
macro avg	0.52	0.40	0.44	2003
weighted avg	0.72	0.76	0.73	2003

Figura 25 - Resultados obtidos (métricas)

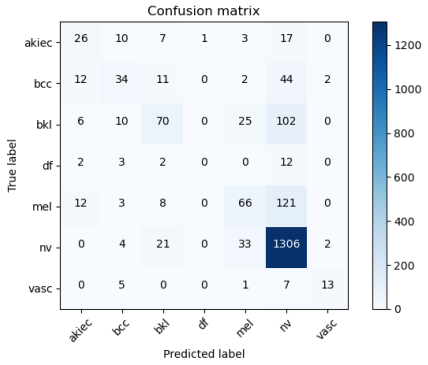

Figura 26 - Matriz confusão

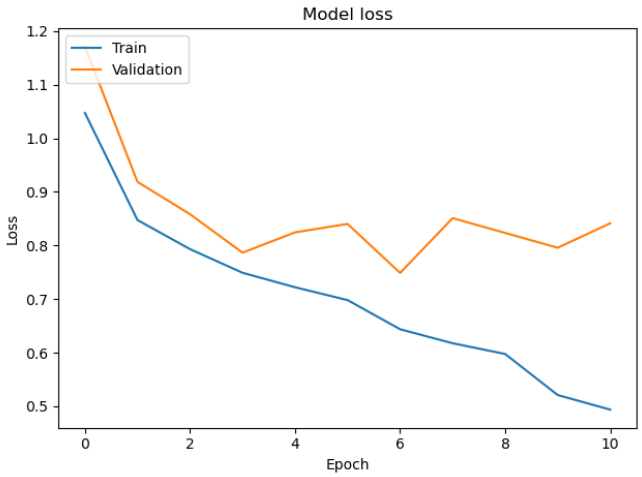

Figura 27 - Variação do custo, ao longo das epochs

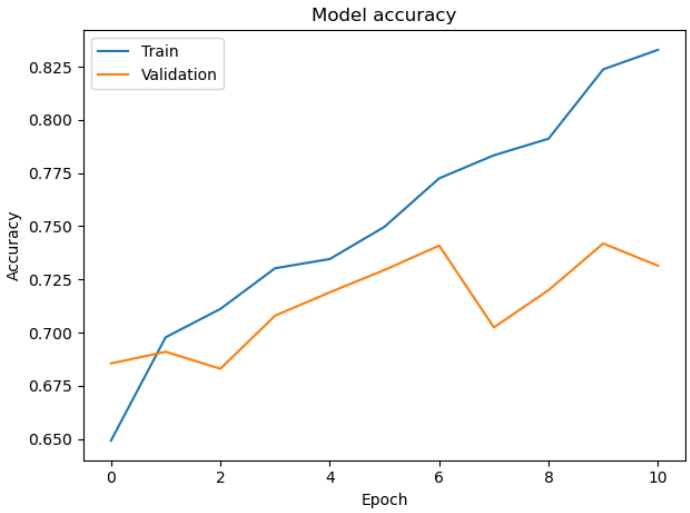

Figura 28 - Variação da accuracy, ao longo das epochs

Observando, as Figuras 25 e 26 é possível comprovar que o aumento da complexidade da rede, não permitiu a obtenção de melhores resultados. Aliás, os resultados revelaram-se inferiores aos obtidos, considerando uma rede menos complexa. O modelo sentiu enormes dificuldades de aprendizagem, algo fácil de constatar, visto que o modelo não conseguiu "acertar" em nenhuma amostra referente à classe *Dermatofibroma*.

Este problema, de dificuldade de aprendizagem, é confirmado através das Figuras ilustradas seguidamente, Figuras 27 e 28. Ambas as Figuras, demonstram a dificuldade de generalização, do modelo. O modelo revelou problemas de *overfitting* muito prematuramente, originando assim os baixos resultados descritos.

A baixa profundidade da rede pode ser um dos problemas, inerentes à dificuldade do modelo em aprender corretamente os dados (para além de outros problemas, como: o baixo nº de dados). A consideração de um nº limitado de camadas convolucionais, implica que as *features* extraídas, sejam "pouco complexas", dificultando assim a identificação de padrões que permitam a classificação correta das imagens, pelas várias classes do problema.

A análise que se segue visa clarificar, se existem vantagens na consideração de redes mais profundas, na resolução deste problema (focando nos resultados obtidos e generalização do modelo).

Variação da profundidade da rede:

O próximo ponto a explorar, enquadra-se na análise do comportamento do modelo, quando se considera a variação do nº de camadas convolucionais da rede.

Ao longo dos anos, foram vários os estudos realizados, com o objetivo de identificar as vantagens e/ou desvantagens, que a adoção de arquiteturas mais profundas oferecem.

Nos últimos anos, foram várias as evidências identificadas, e que comprovam úma maior eficácia e melhores resultados, da aplicação de redes convolucionais mais profundas. Nesse contexto, foram criadas arquiteturas, que exploram essas vantagens, como por exemplo: *Residual Neural Networks (ResNet)*, ou *Inception-V3(4)*.

Da mesma forma, surgiu a necessidade de explorar este fator, sendo definida uma arquitetura mais profunda, face aos exemplos anteriores. Sendo acrescentadas mais duas *Stack Convolutional Layer*, ao modelo.

Para além disso, esta análise compreende a aplicação de dois modelos: um modelo reunindo um baixo nº de parâmetros de treino, e por outro lado um modelo reunindo um maior nº de parâmetros de treino. Foi considerada esta abordagem, de modo a conseguir analisar a "correlação" existente entre a variação da complexidade e da profundidade da rede, em simultâneo. Os melhores resultados identificados nesta fase, irão ser considerados em análises posteriores (existindo naturalmente, outras tentativas pontuais de melhoria dos resultados, mas que serão ligeiras).

Sendo assim, primeiramente irão ser ilustrados os resultados obtidos do aumento da profundidade da rede, considerando um baixo nº de parâmetros. As Figuras 29 e 30 ilustram respetivamente as métricas resultantes da avaliação do modelo, e a matriz confusão. Já, as Figuras 31 e 32 representam a variação do custo e da *accuracy*, ao longo das *epochs*.

akiec	0.17	0.02	0.03	64
bcc	0.52	0.41	0.46	105
bkl	0.40	0.54	0.46	213
df	0.00	0.00	0.00	19
mel	0.75	0.07	0.13	210
nv	0.82	0.96	0.88	1366
vasc	1.00	0.12	0.21	26
accuracy			0.74	2003
macro avg	0.52	0.30	0.31	2003
weighted avg	0.72	0.74	0.69	2003

Figura 29 -Resultados obtidos (métricas)

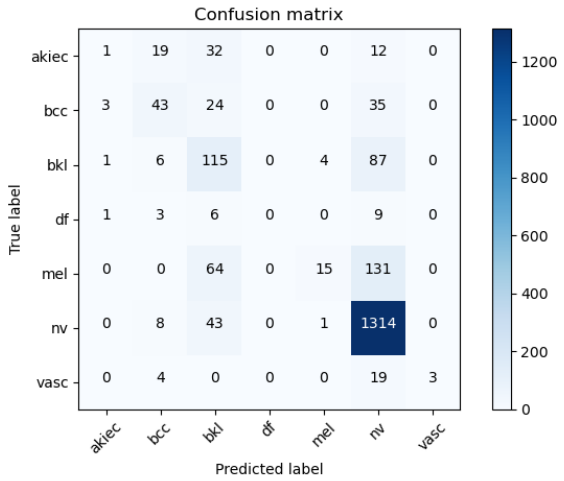

Figura 30 - Matriz confusão

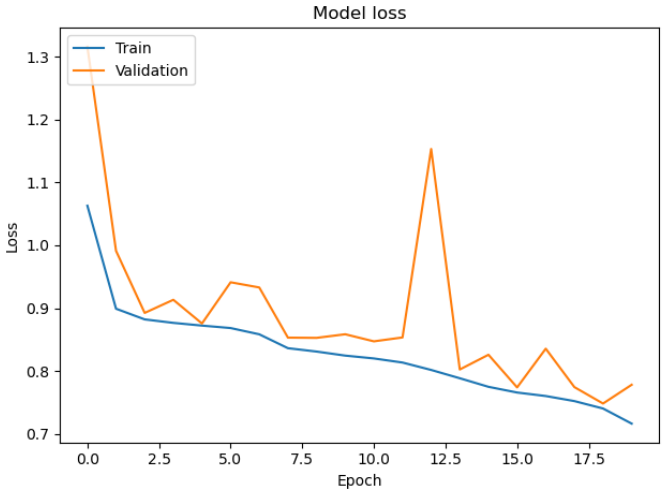

Figura 31 – Variação do custo, ao longo das epochs

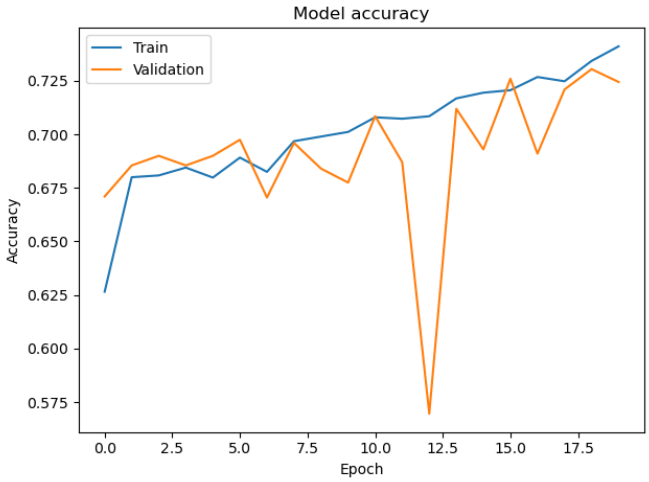

Figura 32 - Variação da accuracy, ao longo das epochs

Apesar dos resultados, ilustrados através das Figuras 29 e 30 ficarem um pouco aquém das expetativas, os mesmos podem ser “enganadores”. Isto é, observando ainda as Figuras 31 e 32

é possível constatar que o modelo ao fim de 20 *epochs*, ainda tem muita margem para aprendizagem, ao contrário do modelo anterior, que ao fim de 5 *epochs* apresentava um custo e uma *accuracy* semelhantes. Para além disso, o modelo ao fim das 20 *epochs* apresentou uma boa capacidade de generalização dos dados (face à aprendizagem em treino), visto que a variação da validação ao longo das *epochs,* é próxima da variação em treino.

Posto isto, o modelo beneficiou do aumento da sua profundidade. É possível ainda concluir, que o modelo, perante o baixo nº de dados atual (10015 imagens), pode apresentar uma maior eficácia considerando uma menor complexidade. Pois, como o nº de dados é baixo, a existência de um elevado nº de parâmetros de treino, pode proporcionar uma baixa generalização dos dados (*overfitting*). O exemplo, que se segue, tenta clarificar esta dúvida.

Segue-se então a análise dos resultados obtidos, considerando um aumento do nº de parâmetros de treino da rede. As Figuras 33 e 34 ilustram respetivamente as métricas resultantes da avaliação do modelo, e a matriz confusão. Já, as Figuras 35 e 36 representam a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.39	0.30	0.34	64
bcc	0.46	0.39	0.42	105
bkl	0.52	0.47	0.50	213
df	0.00	0.00	0.00	19
mel	0.52	0.32	0.40	210
nv	0.84	0.94	0.89	1366
vasc	0.75	0.46	0.57	26
accuracy			0.76	2003
macro avg	0.50	0.41	0.44	2003
weighted avg	0.73	0.76	0.74	2003

Figura 33 - Resultados obtidos (métricas)

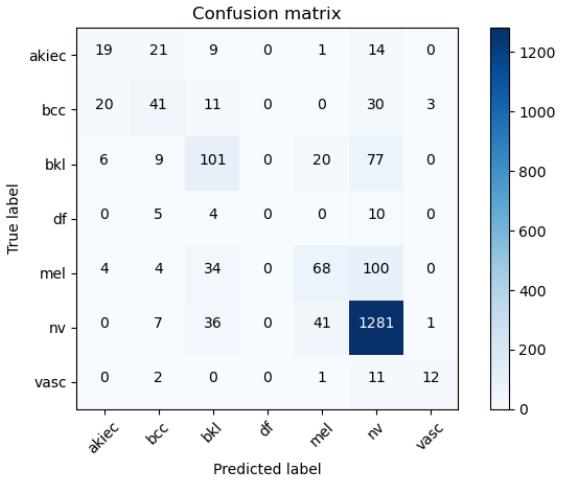

Figura 34 - Matriz Confusão

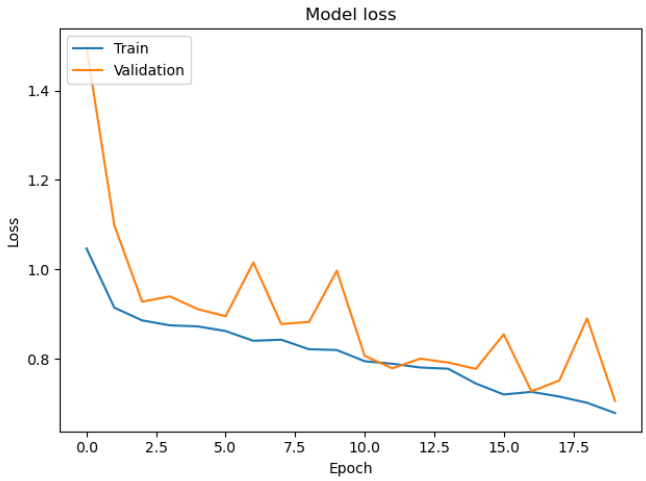

Figura 35 - Variação do custo, ao longo das epochs

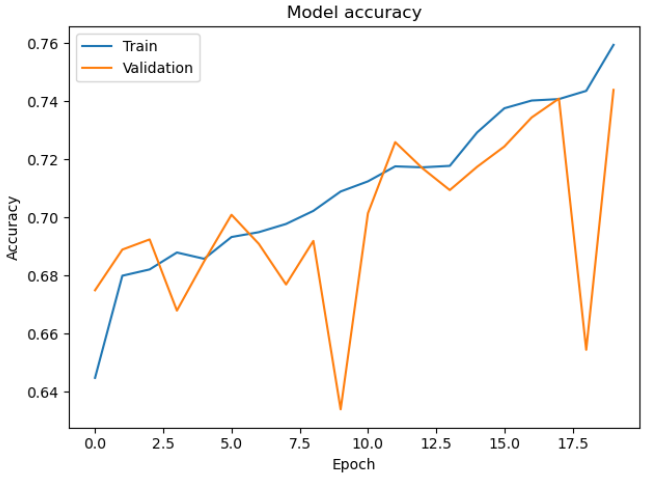

Figura 36 - Variação da accuracy, ao longo das epochs

Recorrendo à análise das Figuras 33 e 34 é possível constatar uma melhoria nos resultados obtidos, apesar de ligeira. Esta melhoria é visível através da métrica *macro avg*, que considera a média ponderada da classificação de cada classe. Dessa forma, é possível constatar que o modelo melhorou a aprendizagem das classes menos balanceadas, como por exemplo: as classes *Vascular skin lesions* (*vasc*), ou *Melanoma* (*mel*). Ainda assim, ao fim de 20 *epochs*, o modelo continua a demonstrar maior aptidão para a classificação das amostras referentes à classe mais representada (*nv*), o que é natural, visto que existem um nº bem maior de amostras relativas a esta classe.

Da observação das Figuras 35 e 36 é possível comprovar que o modelo, ao fim de 20 *epochs*, tal como o modelo anterior ainda se encontra distante da "aprendizagem completa" dos dados, sendo necessário mais treino. O modelo, até ao momento, apresentou bons indicadores quer de aprendizagem quer de generalização. Não apresentando indicadores de possível *overfitting*.

Este modelo revelou indicadores interessantes, mas é necessário recorrer a estratégias de treino, quer permitam melhorar a classificação de todas as classes, e não apenas da classe mais representada. Para além disso é importante aumentar o nº de amostras do problema, de modo a melhorar o processo de aprendizagem do modelo, e também a tornar o processo de aprendizagem mais rápido.

Dessa forma, seguidamente irá ser demonstrada a aplicação de diferentes estratégias de treino (e em simultâneo), com o objetivo de atenuarem as limitações identificadas no parágrafo anterior.

Aplicação de Estratégias de Treino:

De modo a erradicar os vários problemas do *dataset*, e já identificados anteriormente, foram consideradas várias técnicas, de modo a atenuar "estas limitações", com o interesse de melhorar os resultados finais. As estratégias aplicadas foram: *Oversampling*, *Undersampling* e *DataAugmentation*.

As duas primeiras estratégias, tornam o problema "balanceado", considerando abordagens distintas. Já, a *Data Augmentation* atenua outro dos problemas identificados, o baixo nº de amostras.

A aplicação das duas 1ª técnicas, faz apenas sentido considerando *Data Augmentation*, em simultâeneo, de modo a aumentar o nº de amostras do problema (mais precisamente, na aplicação de *Undersampling*, visto que o nº de amostras já apresenta um decréscimo muito substancial).

Adiante, encontram-se descritos os principais resultados obtidos, considerando a utilização das estratégias de treino descritas.

Data Augmentation:

A primeira técnica a considerar é a *Data Augmentation*. Antes de proceder à aplicação das técnicas de ajuste da distribuição das classes (*Undersampling* e *Oversampling*), é importante analisar o comportamento do modelo quando considerando um aumento do nº total de amostras do problema.

Sendo expectável, uma melhoria dos resultados obtidos, visto que geralmente quanto maior o nº de dados considerado, maior a probabilidade do modelo aprender corretamente os dados. Ainda assim, continua a existir uma enorme disparidade, entre o nº de amostras inerentes às várias classes do problema. Dificultando assim, a aprendizagem e generalização do modelo.

Posto isto, foi aplicado o modelo mais ajustado ao problema, e que fora identificado durante esta análise (elevada profundidade e baixa/média complexidade), considerando a adoção de *Data Augmentation*.

As Figuras 37 e 38 ilustram respetivamente as métricas resultantes da avaliação do modelo, e a matriz confusão. Já, as Figuras 39 e 40 representam a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.37	0.30	0.33	64
bcc	0.47	0.56	0.51	105
bkl	0.59	0.33	0.43	213
df	0.00	0.00	0.00	19
mel	0.51	0.26	0.35	210
nv	0.82	0.96	0.88	1366
vasc	0.80	0.31	0.44	26
accuracy			0.76	2003
macro avg	0.51	0.39	0.42	2003
weighted avg	0.72	0.76	0.73	2003

Figura 37 - Resultados obtidos (métricas)

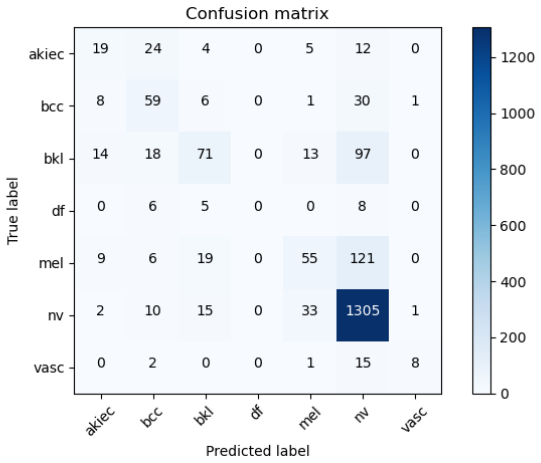

Figura 38 - Matriz Confusão

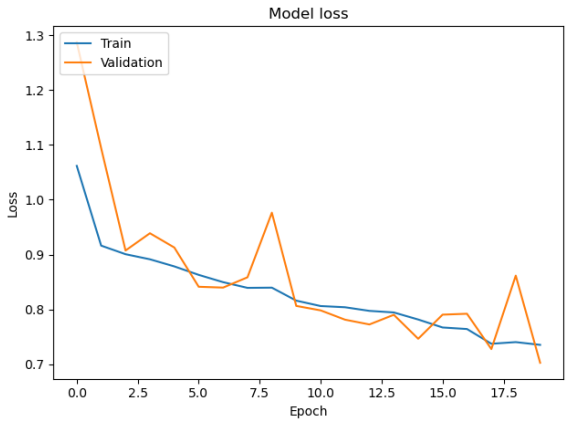

Figura 39 - Variação do custo, ao longo das epochs

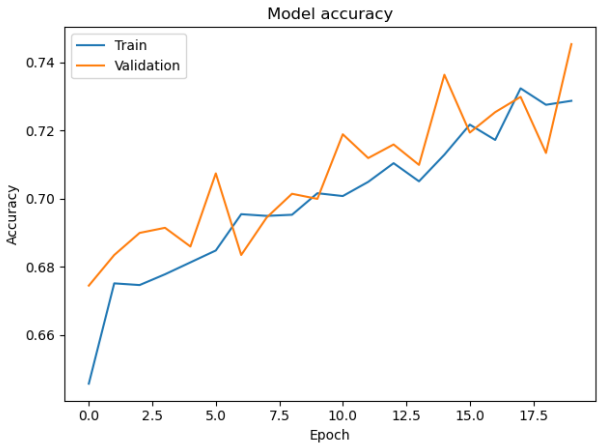

Figura 40 - Variação da accuracy, ao longo das epochs

Os resultados, ilustrados através das Figuras 37 e 38, foram semelhantes aos obtidos previamente. Ou seja, a inclusão de um maior nº de amostras, não permitiu ao modelo melhorar a sua aprendizagem, nem os resultados.

Tal, como já tinha sido indicado atrás, a inclusão única de *Data Augmentation* não permite a resolução da maior limitação do *dataset*: a distribuição inadequada do nº de amostras, pelas classes do problema.

Sendo assim, era expectável que os resultados fossem semelhantes, comparando com a sua não utilização (*Data Augmentation*). Ainda assim, estava prevista uma melhoria ligeira dos resultados obtidos, até aqui.

Como podemos constatar, novamente através das Figuras 37 e 38, o modelo limita-se a aprender os dados referentes à classe mais representada, não conseguindo identificar padrões que lhe permitam classificar corretamente as amostras referentes às restantes classes. Ou seja, é muito importante garantir, que as classes do problema sejam balanceadas, de modo a tentar atenuar a dificuldade que o modelo apresenta na classificação correta de todas as classes.

Analisando as Figuras 39 e 40, é possível identificar que o modelo apresentou uma correta generalização dos dados. Contudo, o modelo revelou uma aprendizagem muito lenta. Este facto pode estar diretamente relacionado com o hiperparâmetro *learning rate*, existindo a necessidade de aumentar um pouco mais, ou o modelo necessita de mais complexidade, isto é, do aumento do seu nº de hiperparâmetros.

Mas, resumidamente é necessário seguidamente dar importância à utilização conjunta das técnicas *Oversampling* e *Data Augmentation*.

***Undersampling* e *Data Augmentation*:**

A primeira técnica de ajuste da distribuição das classes utilizada é: *Undersampling*. Como o nº total de amostras é baixo, a sua aplicação não é tão plausível como a aplicação de *Oversampling*. Contudo, a sua aplicação em simultâneo com *Data Augmentation*, pode revelar-se benéfica. Visto que, para além de se ajustar o nº de amostras de uma forma equitativa, são introduzidas variações das amostras já existentes.

Apesar, de considerada a utilização de *Data Augmentation* continua a existir um nº baixo de amostras, dificultando assim a aprendizagem do modelo.

Ainda assim, foi considerada a utilização conjunta de ambas as estratégias. Da sua aplicação resultaram os resultados ilustrados na Figura 41, e a Matriz Confusão apresentada através da Figura 42. Já, as Figuras 43 e 44 descrevem respetivamente a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.00	0.00	0.00	64
bcc	0.50	0.18	0.27	105
bkl	0.49	0.23	0.31	213
df	0.00	0.00	0.00	19
mel	0.00	0.00	0.00	210
nv	0.71	0.98	0.83	1366
vasc	0.00	0.00	0.00	26
accuracy			0.70	2003
macro avg	0.24	0.20	0.20	2003
weighted avg	0.57	0.70	0.61	2003

Figura 41- Resultados obtidos (métricas)

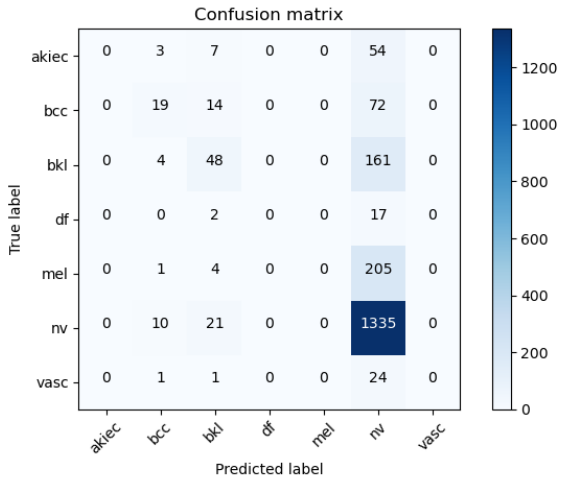

Figura 42 - Matriz Confusão

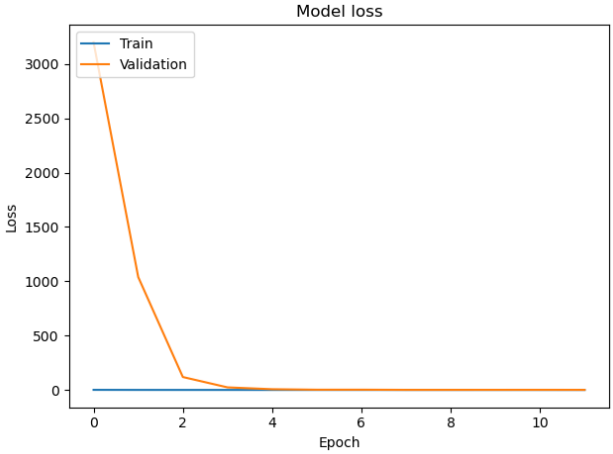

Figura 43 - Variação do custo, ao longo das epochs

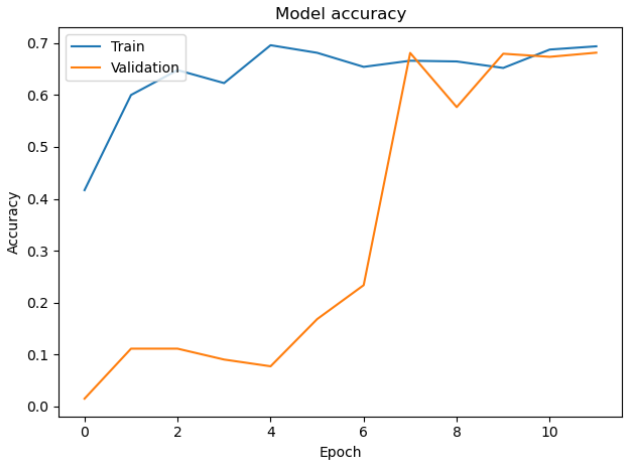

Figura 44 - Variação da accuracy, ao longo das epochs

Tal como já anunciado, os resultados obtidos ficaram muito aquém do desejado. Aliás os resultados revelaram-se extremamente baixos, tal como podemos observar através das Figuras 41 e 42.

Apesar de se ter erradicado o problema do não balanceamento das classes, o nº de amostras utilizados decresceu muito significativamente, visto que a distribuição do nº de amostras pelas classes era muito heterogénea. A inclusão de *Data Augmentation* nesta situação, não revela benefícios (se revelar, são quase nulos), dada a limitação existente no nº de amostras.

A consideração de um baixo nº de amostras dificultou o processo de aprendizagem do modelo, visível através das Figuras 43 e 44. O modelo estagnou precocemente, não conseguindo extrair mais conhecimento dos dados. Algo que, está diretamente ligado ao baixo nº de amostras consideradas.

Seguidamente, irá ser demonstrada a aplicação do modelo considerada outra técnica de ajuste da distribuição das classes, *Oversampling*. Esta técnica, perante o problema em questão, já se revela mais adequada, e certamente permitirá a obtenção de melhores resultados.

Oversampling* e *Data Augmentation:

Finalmente, a última estratégia de treino a aplicar é *Oversampling*. Tendo em consideração o problema, esta estratégia revela-se mais plausível do que a utilização da técnica *Undersampling*, visto que, o nº total de amostras já é reduzido.

Desta forma, a aplicação desta estratégia permite o ajuste do nº de amostras pelas várias classes, através da duplicação de amostras, nas classes menos balanceadas. Esta abordagem atenua o problema da "não" distribuição das classes, mas a duplicação das imagens, não permite aos modelos a extração de "novas *features*". Sendo assim, e de modo a permitir ao modelo, a eventual aprendizagem/identificação de particularidades nos dados, foi considerada a utilização em simultâneo da técnica *Data Augmentation*. A introdução de pequenas modificações nos dados, pode ajudar os modelos na identificação correta das amostras, ainda que não seja expectável uma melhoria acentuada, mas sim ligeira.

As Figuras 45 e 46 ilustram respetivamente os resultados obtidos e a Matriz Confusão. Já, as Figuras 47 e 48 sintetizam a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.40	0.30	0.34	64
bcc	0.64	0.47	0.54	105
bkl	0.39	0.66	0.49	213
df	0.00	0.00	0.00	19
mel	0.57	0.06	0.11	210
nv	0.86	0.92	0.89	1366
vasc	0.88	0.58	0.70	26
accuracy			0.75	2003
macro avg	0.53	0.43	0.44	2003
weighted avg	0.74	0.75	0.72	2003

Figura 45 - Resultados obtidos (métricas)

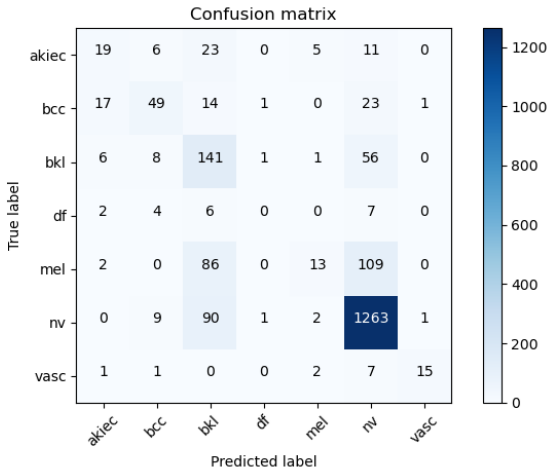

Figura 46 - Matriz Confusão

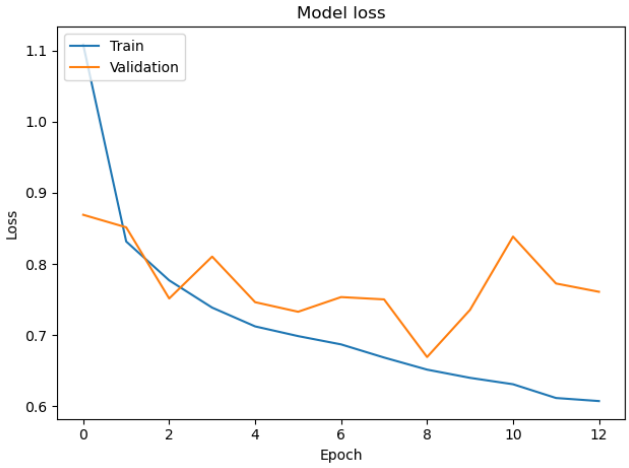

Figura 47 - Variação do custo, ao longo das epochs

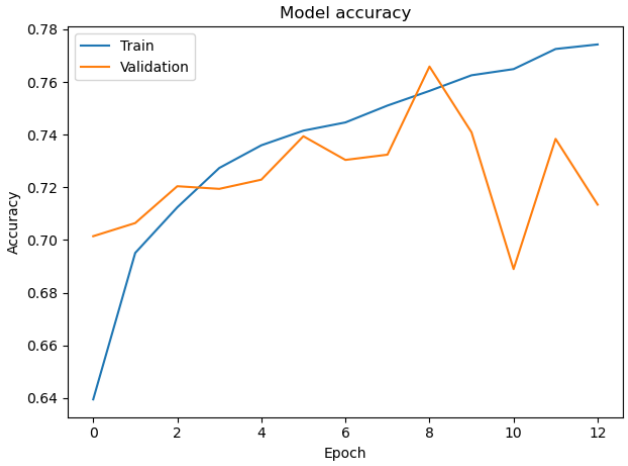

Figura 48 - Variação da accuracy, ao longo das epochs

Os resultados obtidos ficaram um pouco aquém das expetativas. Ou seja, a introdução de mais amostras da aplicação de ambas as técnicas, e o ajuste da distribuição das classes, não promoveu a obtenção de melhorias assinaláveis nos resultados finais. As melhorias foram

ligeiras, como podemos observar através das Figuras 45 e 46, onde a *macro accuracy* apresentou apenas uma ligeira subida, face aos resultados anteriores (não aplicação de ambas as técnicas).

É difícil identificar os motivos que estão por detrás deste facto. Mas, a contínua aprendizagem apenas da classe mais representada, sugere que a introdução das imagens duplicadas (*Oversampling)*, e de novas variantes das imagens (*Data Augmentation*), não permitiu aos modelos a extração de características mais complexas dos dados, e que permitam a classificação correta de todas as classes do problema. Dessa forma, o modelo manteve a dificuldade inicial de classificação das classes menos representativas.

Comparando, esta abordagem com as abordagens anteriores, isto é, sem a utilização deste tipo de técnicas, é visível que os resultados são semelhantes, não existindo assim benefícios significativos na sua utilização. Contudo, é necessário ter em consideração que o modelo revelou problemas de *overfitting* a partir da *epoch* 8, visível através das Figuras 47 e 48. Isto, permite constatar que o modelo ainda não se encontra totalmente otimizando, podendo assim concluir que os resultados finais ainda possam ser melhorados. Sendo, necessário proceder à sua otimização, de modo a retirar conclusões mais assertivas desta análise.

Segue-se assim, a análise de outra arquitetura CNN, a *VGGNet*.

VGGNet:

A próxima rede *CNN* em análise, é a rede *VGGNet*. Esta arquitetura é mais recente que a *AlexNet*, sendo da mesma forma considerada uma rede mais robusta, e mais eficaz. A rede *VGGNet* é utilizada em larga escala pelos *datascientists*, devido sobretudo à sua facilidade de interpretação e de construção, para além dos bons resultados obtidos.

Sendo assim, foi considerada a criação de uma rede baseada na sua arquitetura, mais concretamente a rede *VGGNet – 16*.

Apesar de ser uma rede bastante eficiente, revela algumas limitações, como: a sua elevada exigência computacional. Isto é, a utilização de *stacked convolutional layers* profundas, aumenta significativamente o nº de parâmetros de treino da rede, exigindo assim um maior poder computacional, e mais tempo de treino. Ou seja, é necessário ter este ponto em consideração, antes de definir o nº de filtros das redes convolucionais.

É difícil prever os resultados, que esta arquitetura irá fornecer ao problema, visto que são várias as condicionantes em causa, ainda assim é expectável que os resultados não sejam inferiores ao modelo *AlexNet*.

A análise desta arquitetura irá ser mais simples, comparativamente à rede *AlexNet*, visto que as condicionantes consideradas anteriormente, aplicam-se na maioria dos casos também a esta rede, como por exemplo: benefícios de incorporação de redes mais complexas ou de redes mais profundas, dado que o problema a resolver é o mesmo. Sendo assim, a documentação dos vários casos aplicados, irá ser reduzida.

Análise à complexidade da rede:

Ao contrário do *AlexNet*, não irá ser efetuada uma análise exaustiva deste ponto, visto que o *hardware* à disposição não permite a aplicação de redes muito complexas, visto que facilmente esta rede extrapola um nº muito elevado de parâmetros de treino.

Dessa forma, irá ser considerada uma rede *VGGNet standard*, que reúne uma complexidade ponderada, e de acordo com a complexidade do problema.

Para tal, foram realizados vários testes à rede (ligeiros, dado que a otimização do modelo irá ser abordada mais adiante), de modo a identificar o valor mais adequado de filtros das várias redes convolucionais (tendo em consideração o problema a resolver).

Os exemplos descritos posteriormente consideram a arquitetura identificada anteriormente.

Análise à profundidade da rede:

De modo a aumentar a flexibilidade dos modelos e ainda de modo a reduzir a carga computacional deste tipo de rede, é necessário adotar arquiteturas de média profundidade. Caso contrário, o nº de parâmetros de treino torna-se incomportável. Sendo que para além disso, o modelo não consegue extrair *features* mais complexas, limitando-se a identificar características mais superficiais, o que pode prejudicar a sua *performance*. Também não se podem considerar redes muito profundas, senão a aprendizagem torna-se muito lenta.

Sendo assim, foi considerada a utilização de uma rede *VGGNet* composta por um nº considerável de *stacked convolutional layers*, mais concretamente: 4 camadas.

Foi possível constatar através da aplicação desta rede, que a inclusão de um maior nº de camadas permite ao utilizador flexibilizar da melhor forma a rede, tendo em conta o problema a resolver. Ou seja, o utilizador consegue desta forma criar uma rede que se ajusta o mais possível, quer ao nº de dados do problema, quer à complexidade do mesmo. Algo, que é muito difícil de concretizar, considerando uma *shallow VGGNet*.

Posto isto, o autor recorreu à aplicação desta rede. Para já não considerando a aplicação de qualquer estratégia, de aumento do nº de dados, ou ainda para ajustar a distribuição das amostras, ao longo das classes.

As Figuras 49 e 50 ilustram respetivamente as métricas obtidas e ainda a Matriz Confusão. Já, as Figuras 51 e 52 descrevem a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.30	0.05	0.08	64
bcc	0.52	0.50	0.51	105
bkl	0.36	0.67	0.47	213
df	0.00	0.00	0.00	19
mel	0.85	0.05	0.10	210
nv	0.84	0.90	0.87	1366
vasc	0.35	0.31	0.33	26
accuracy			0.72	2003
macro avg	0.46	0.35	0.34	2003
weighted avg	0.74	0.72	0.69	2003

Figura 49 - Resultados obtidos (métricas)

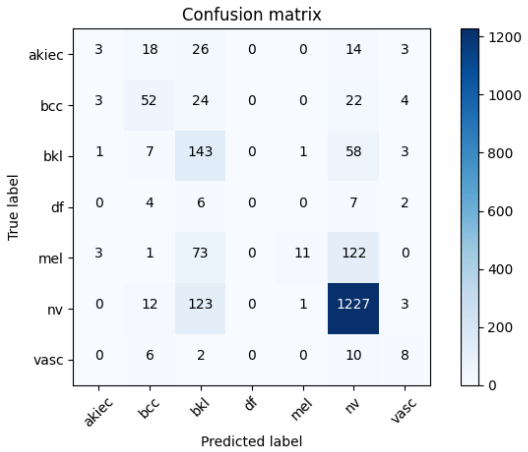

Figura 50 - Matriz Confusão

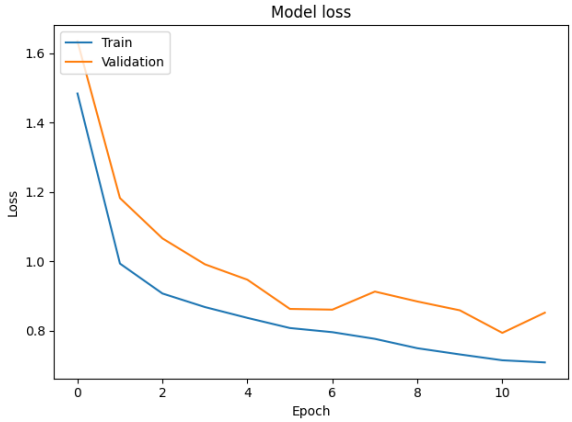

Figura 51 - Variação do custo, ao longo das epochs

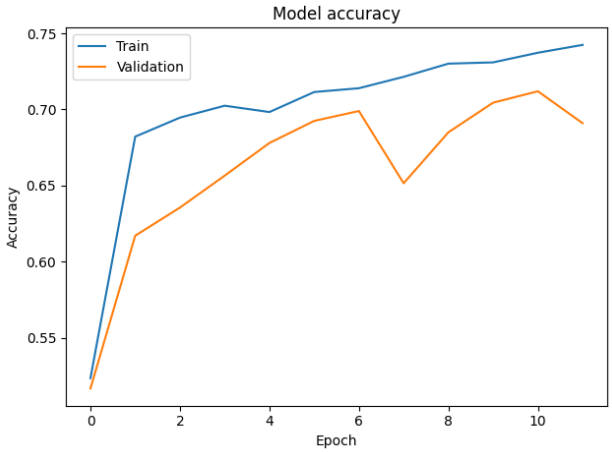

Figura 52 - Variação da accuracy, ao longo das epochs

Os resultados obtidos ficaram um pouco abaixo do esperado, sendo inclusive inferiores aos resultados da aplicação da arquitetura *ResNet*. Ou seja, através da visualização das Figuras 49 e 50 é possível concluir que o modelo apresentou inúmeras dificuldades na classificação das amostras referentes às 6 classes menos representativas do problema.

Esta baixa capacidade do modelo, está subjacente nas Figuras 51 e 52. Isto é, o modelo ao fim das 12 *epochs* consideradas, encontra-se ainda longe de dar por terminado o processo de treino. Existindo assim ainda bastante margem de progressão para melhorar os resultados. Sendo que, seria importante otimizar o modelo, de modo a melhorar a sua generalização. Para além disso, seria importante aumentar a velocidade de treino do modelo, porque o mesmo está a convergir lentamente. Este problema, pode eventualmente estar relacionado com o facto de existirem poucas amostras de treino, sendo necessário recorrer a abordagens para aumentar esse número.

Seguidamente, e com o objetivo de melhorar os resultados obtidos, recorreu-se à aplicação do modelo, considerando a utilização das técnicas *Oversampling* e *Data Augmentation*. Sendo expectável uma melhoria dos resultados obtidos, esperando-se ao contrário do que acontecera com a rede *AlexNet*, uma melhoria significativa dos resultados.

Oversampling + Data Augmentation:

Relativamente às estratégias de treino a aplicar, irá ser apenas considerada a aplicação simultâneo de *Oversampling* e *Data Augmentation*.

Como fora demonstrado, através da análise à rede *AlexNet*, a única abordagem que faz sentido aplicar neste problema, é a aplicação simultânea das duas estratégias citadas no parágrafo anterior.

Desse modo, foi considerada a aplicação conjunta de ambas as estratégias, com o objetivo de perceber quais as vantagens e/ou desvantagens, associadas à sua utilização.

As Figuras 53 e 54 ilustram respetivamente as métricas obtidas e a Matriz Confusão. Já, as Figuras 55 e 56 descrevem a variação custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.49	0.28	0.36	64
bcc	0.61	0.64	0.62	105
bkl	0.59	0.44	0.50	213
df	0.30	0.16	0.21	19
mel	0.61	0.33	0.43	210
nv	0.85	0.96	0.90	1366
vasc	0.67	0.69	0.68	26
accuracy			0.79	2003
macro avg	0.59	0.50	0.53	2003
weighted avg	0.77	0.79	0.77	2003

Figura 53 - Resultados obtidos (métricas)

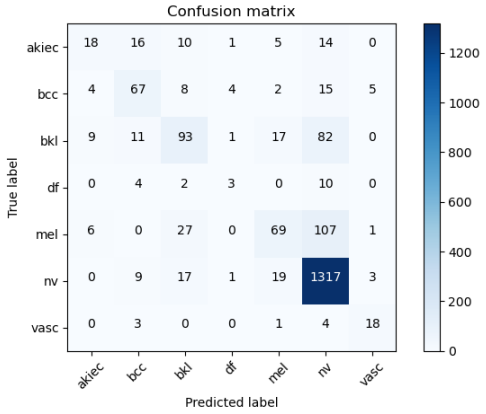

Figura 54 - Matriz Confusão

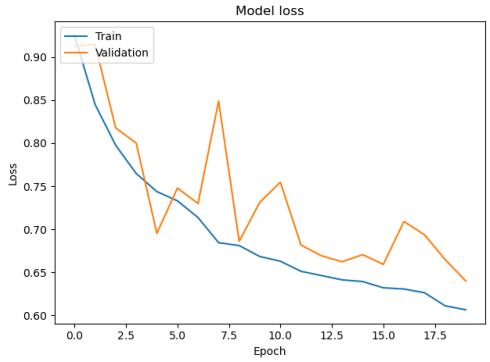

Figura 55 - Variação do custo, ao longo das epochs

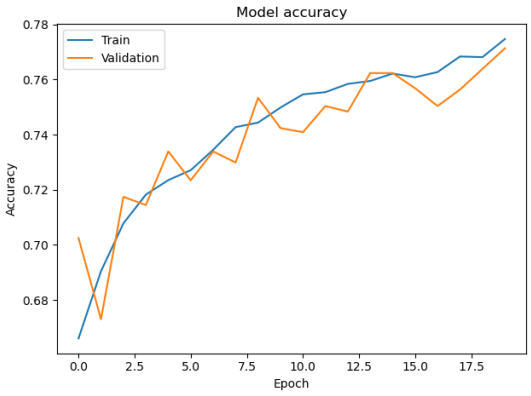

Figura 56 - Variação da accuracy, ao longo das epochs

A aplicação de ambas as técnicas em simultâneo, originou uma melhoria significativa nas métricas, como podemos observar através da Figura 53. Recorrendo novamente à observação da Figura 53 é possível constatar, que na globalidade de todos os exemplos descritos até aqui, este modelo foi o que revelou uma classificação das amostras mais uniforme. Ou seja, o modelo conseguiu extrair *features* mais relevantes dos dados, o que lhe permitiu acertar um maior nº de amostras, tendo em conta todas as classes e não só a classe mais representativa (*nv*). Este facto, é comprovado através da métrica *macro average*. O modelo reuniu um *f-score* positivo, algo que até aqui, nunca fora atingido. Revelando assim uma melhoria visível do processo de aprendizagem e de generalização do modelo. A Figura 54 destaca uma maior quantidade de acertos, por parte do modelo, face aos modelos descritos ao longo do estudo.

As Figuras 55 e 56 demonstram uma generalização adequada do modelo, ao longo das 20 *epochs* consideradas. Esclarecendo ainda a existência de "margem de manobra", por parte do modelo, para melhorar a sua aprendizagem. Visto que o modelo ainda se encontra longe de garantir uma aprendizagem completa dos dados. Mas, recorrendo à observação de ambas as Figuras é possível destacar que o modelo apresenta uma aprendizagem lenta, dando a entender que a rede não é profunda o suficiente, para que seja possível concluir e extrair toda a informação relativa aos dados. Sendo importante, observar o comportamento do modelo, quando confrontando com uma arquitetura mais profunda.

Tal como expectável, o modelo *VGGNet* demonstrou uma melhoria nos resultados obtidos, face à arquitetura *AlexNet*. A maior complexidade da rede *VGGNet* permite a extração de *features* mais relevantes para o problema. Contudo, é necessário ter em consideração que esta rede exige um maior poder computacional. Mas, na globabilidade esta rede revelou-se mais capaz de extrair conhecimento dos dados.

Seguidamente, irá ser considerada uma rede ainda mais complexa, do que a rede *VGGNet*, a rede *ResNet.* É uma rede bastante flexível e bastante complexa. Em diversos estudos, revela-se a arquitetura que apresenta melhores resultados. Sendo, da mesma forma expectável a obtenção de ainda melhores resultados.

***ResNet**:*

Para além da aplicação da rede *AlexNet* e *VGGNet*, foi considerada a utilização da rede *ResNet*.

Esta rede é a mais recente das três redes, sendo também a rede mais complexa de entender e de aplicar. Contudo, têm-se demonstrado uma rede muito eficaz na resolução dos mais variados tipos de problema. Sendo, nos dias de hoje ponderada e aplicada em qualquer problema ou *benchmark*.

A principal característica deste tipo de rede, é a sua elevada profundidade. Isto é, é possível criar arquiteturas baseadas em *ResNet,* com centenas de camadas convolucionais, como por exemplo: *Resnet-101.*

Contudo, é necessário ter em consideração quer este tipo de rede requer um elevado poder computacional, na sua manipulação. Sendo assim, irão ser consideradas abordagens menos profundas deste tipo de rede, e baseadas na rede *ResNet-18*.

Seguidamente, irão ser expostas duas abordagens da sua aplicação: (1) aplicação sem recorrer a quaisquer técnicas de aumento do nº de dados e de ajuste da distribuição das classes, e (2) utilização das técnicas *Oversampling* e *Data Augmentation*.

Aplicação sem qualquer técnica:

O exemplo descrito seguidamente, representa o *output* obtido, considerando uma série de exemplos aplicados, na tentativa de encontrar uma rede adequada para o problema (procura não exaustiva, visto que essa tarefa, destina-se ao PSO).

Essa procura, teve em consideração várias condicionantes, como a complexidade da rede, ou a sua profundidade. Dadas as limitações e *hardware* existentes, existiu uma ligeira restrição no aumento da complexidade do problema, mas que em nada afetou os resultados finais obtidos.

Dessa forma, as Figuras ilustradas seguidamente demonstram os resultados obtidos. As Figuras 57 e 58 salientam os resultados das métricas de avaliação consideradas e ainda a Matriz Confusão. Já as Figuras 59 e 60 espelham a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.42	0.20	0.27	64
bcc	0.46	0.53	0.49	105
bkl	0.62	0.34	0.44	213
df	0.33	0.05	0.09	19
mel	0.61	0.14	0.23	210
nv	0.80	0.97	0.88	1366
vasc	0.91	0.38	0.54	26
accuracy			0.75	2003
macro avg	0.59	0.38	0.42	2003
weighted avg	0.73	0.75	0.71	2003

Figura 57 - Resultados obtidos (métricas)

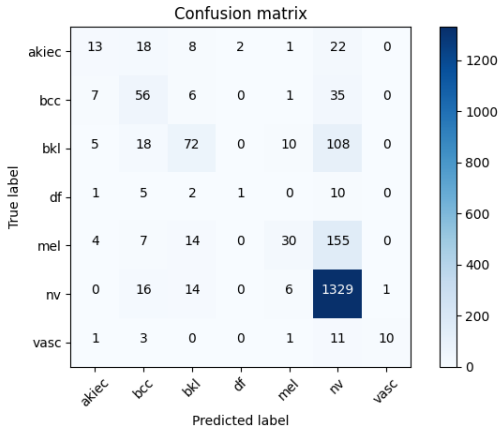

Figura 58 - Matriz Confusão

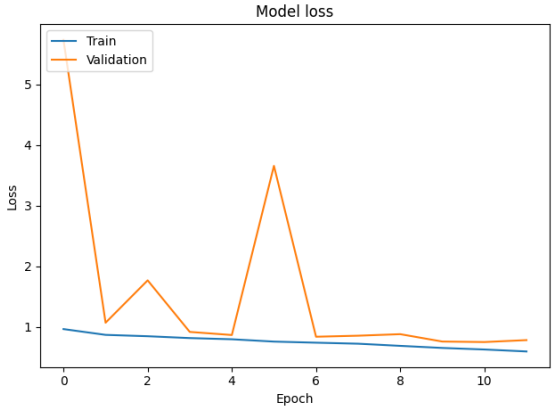

Figura 59 - Variação do custo, ao longo das epochs

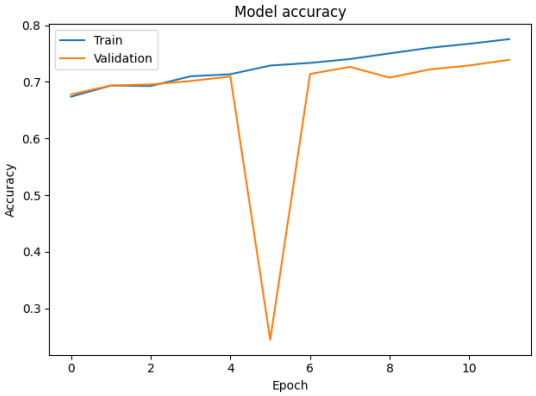

Figura 60 - Variação da accuracy, ao longo das epochs

Os resultados obtidos, apesar de não serem muito expressivos, ainda assim foram superiores, aos obtidos, considerando a aplicação da rede *AlexNet* e *VGGNet* (comparando abordagens similares – isto é, não considerando a aplicação de *Oversampling* e *Data Augmentation*).

Observando a Figura 57 é possível comprovar que o modelo para além de apresentar uma *accuracy* superior, na ordem dos 75%, revela um *f-score* superior, em praticamente todas as classes do problema. Podendo assim concluir, que o modelo revelou uma melhor aprendizagem dos dados.

Analisando as Figuras 59 e 60 podemos observar uma adequada generalização do modelo. Eventualmente, o modelo tenderia a obter melhores resultados, contudo e devido à utilização de *callbacks* (controlam o custo e a *accuracy* em treino – *early stopping*), o treino terminou mais cedo do que o previsto. Contudo caso, o treino se estendesse seria expectável uma melhoria nos resultados finais.

Tal como aconteceu com a rede *VGGNet,* é provável que a utilização das técnicas *Oversampling* e *Data Augmentation*, possam ajudar o modelo a extrair mais conhecimento dos dados, representando assim melhores resultados. Seguidamente, recorreu-se à aplicação de ambas as técnicas.

Oversampling + Data Augmentation:

Tal como fora demonstrado para os exemplos anteriores, a arquitetura *ResNet* também irá ser aplicada considerando a utilização das técnicas *Oversampling* e *Data Augmentation*.

O objetivo é exatamente o mesmo, verificar se existem ou não vantagens associadas à sua aplicação, mais concretamente o aumento do nº de amostras e ainda do ajuste da distribuição das classes.

Segue-se então a demonstração dos resultados obtidos, visíveis através das Figuras 61 e 62. Já, as Figuras 63 e 64 descrevem a variação do custo e da *accuracy*, ao longo das *epochs*.

	precision	recall	f1-score	support
akiec	0.46	0.52	0.49	64
bcc	0.54	0.60	0.57	105
bkl	0.61	0.39	0.48	213
df	0.25	0.37	0.30	19
mel	0.42	0.65	0.51	210
nv	0.91	0.87	0.89	1366
vasc	0.89	0.62	0.73	26
accuracy			0.76	2003
macro avg	0.58	0.57	0.57	2003
weighted avg	0.79	0.76	0.77	2003

Figura 61 - Resultados obtidos (Métricas)

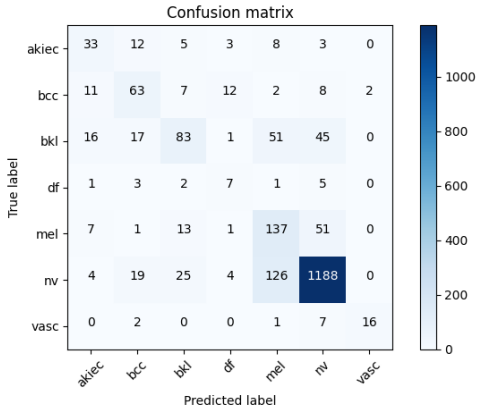

Figura 62 - Matriz Confusão

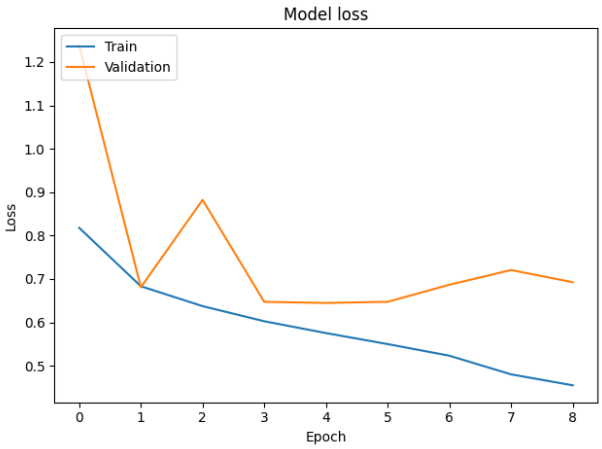

Figura 63 - Variação do custo, ao longo das epochs

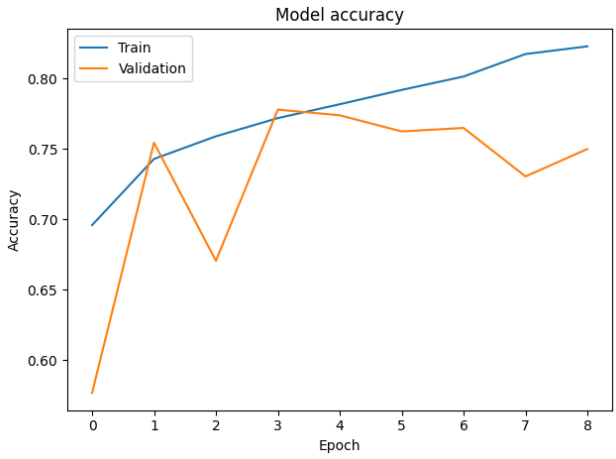

Figura 64 - Variação da accuracy, ao longo das epochs

Recorrendo à observação das Figuras 61 e 62, podemos concluir que a aplicação conjunta de ambas as técnicas, permitiu a obtenção de resultados satisfatórios, apesar de ainda estarem longo do pretendido, mas tendo em conta todas as condicionantes em causa, os resultados foram interessantes.

A métrica *macro average*, encontra-se mais otimizada, demonstrando assim que a classificação das várias classes do problema, é mais uniforme. Isto é, o modelo já apresenta uma melhor aprendizagem, conseguindo demonstrar que consegue classificar corretamente (na ordem dos 50%), todas as classes (excetuando a classe *df*).

Ou seja, é possível concluir que a introdução de um maior nº de amostras no problema permitiu ao modelo aprender, de uma forma mais correta os dados. O ajuste da distribuição das classes, muito provavelmente também permitiu ao modelo, melhorar os resultados obtidos.

Contudo, e observando as Figuras 63 e 64 é possível constatar que o modelo tende a estagnar muito prematuramente (generalização). Impedindo assim a obtenção de resultados ainda melhores. Da análise a ambos os gráficos, é possível concluir que existe ainda margem de manobra para melhorar os resultados obtidos, sendo assim necessário recorrer à otimização do modelo, para tal.

Numa primeira fase, e de modo a tentar melhorar os resultados obtidos, irá ser criado um *Ensemble*, que irá agregar os três melhores modelos obtidos nesta análise, isto é, o melhor modelo identificado, para cada uma das arquiteturas analisadas. Para tal, irá ser considerada a média dos seus pesos.

Mais adiante, e com a ajuda do algoritmo de otimização *Particle Swarm Optimization*, irá ser efetuada a otimização de alguns dos parâmetros deste modelo, com o intuito de melhorar os resultados obtidos, e ao mesmo tempo reduzir a carga computacional do modelo.

***Ensemble**:*

Descritas as várias arquiteturas consideradas na resolução do problema, segue-se a aplicação da técnica *ensemble*, considerando os modelos obtidos.

A técnica *Ensemble*, considera o agrupamento das previsões de vários modelos, com o objetivo de reduzir a variância e melhorar os resultados obtidos.

Ou seja, os *outputs* inerentes aos vários modelos considerados (*AlexNet, VGGNet* e *ResNet)*, serão combinados. Para tal irá ser considerada a média das suas previsões.

A aplicação desta técnica reduz a variância, visto que, não é apenas considerado o *output* único de um modelo, mas sim de três modelos.

Em muitos casos, ajuda também a melhorar os resultados. Contudo, é muito difícil prever o impacto da adoção desta técnica nas métricas. Uma vez que, a melhoria está dependente da estabilidade dos modelos. Isto é, caso os modelos sejam instáveis, então certamente os resultados irão melhorar, caso contrário podem diminuir.

Sendo assim, procedeu-se à combinação dos três "melhores" modelos identificados atrás (um modelo, referente a cada uma das arquiteturas descritas). Tal como já fora referido, foi combinada a média das previsões dos três modelos.

As Figuras 65 e 66 ilustram os resultados obtidos e a matriz confusão resultante, da aplicação da técnica *ensemble*.

	precision	recall	f1-score	support
akiec	0.49	0.48	0.49	64
bcc	0.69	0.56	0.62	105
bkl	0.56	0.60	0.58	213
df	0.45	0.26	0.33	19
mel	0.62	0.31	0.42	210
nv	0.88	0.95	0.91	1366
vasc	0.91	0.81	0.86	26
accuracy			0.80	2003
macro avg	0.66	0.57	0.60	2003
weighted avg	0.79	0.80	0.79	2003

Figura 65 - Resultados obtidos (métricas)

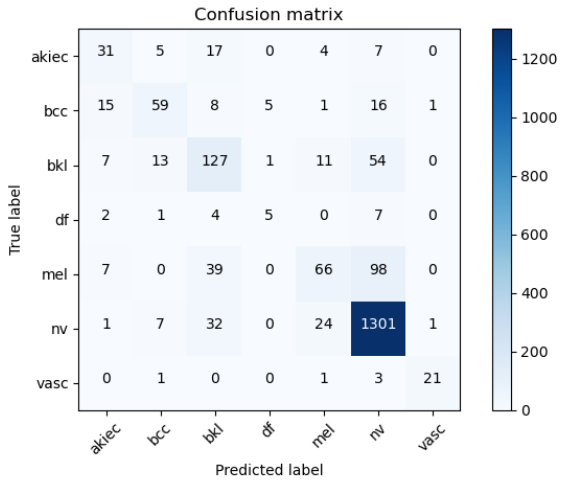

Figura 66 - Matriz Confusão

Observando a Figura 65 é possível observar uma melhoria global das métricas. Sendo possível atingir a meta dos 60% na métrica *f-score macro average*.

Estes resultados eventualmente poderiam ser mais expressivos, uma vez que foi considerado um nº baixo de *epochs* no treino dos três modelos considerados. Caso, o nº de *epochs* fosse superior, muito provavelmente os resultados seriam superiores.

Recorrendo à observação da Figura 66, é possível constatar que continua a existir uma maior dificuldade do modelo, na previsão das amostras referentes às classes *Melanocytic nevi (nv)* e *Melanoma (mel)*.

A aplicação da técnica *Ensemble* permitiu evidenciar as suas principais vantagens. Visto que, para além de ter permitido a obtenção de resultados mais aprimorados, reduziu a variância dos modelos, aumentando assim a "veracidade" dos modelos criados.

Esta técnica irá ser utilizada mais adiante, e aplicada com maior robustez e com maior cuidado. Isto é, o *PSO* irá ser aplicado, na otimização das três arquiteturas definidas e descritas atrás. Após a identificação dos hiperparâmetros mais adequados para os modelos, prossegue-se para a tarefa de treino dos modelos, sendo considerado um processo mais eficiente, contando com um maior nº de *epochs*, ajudando assim a melhorar a aprendizagem dos modelos. Em última instância, recorre-se ao agrupamento/combinação dos três modelos otimizados pelo *PSO*, e já devidamente treinados. Sendo expectável que a aplicação do *PSO,* a par da técnica *ensemble* permitam a obtenção de resultados ainda melhores.

Particle Swarm Optimization:

Finalmente, e depois de concluída toda a análise do problema, bem como a aplicação de diferentes arquiteturas *CNN*, segue-se a otimização dos modelos exemplificados anteriormente.

O algoritmo de otimização a aplicar é o *Particle Swarm Optimization*. O seu objetivo baseia-se na otimização de alguns dos hiperparâmetros dos modelos, e de variáveis relacionadas com o processo de treino, como: o nº de filtros das redes convolucionais, ou o *batch size*. Com o objetivo de identificar um modelo mais robusto, mais rápido, menos exigente computacionalmente, e ao mesmo tempo garantindo resultados similares, ou melhores.

É expectável a obtenção de modelos bem mais eficientes, e demonstrando melhores métricas. Visto que, a análise efetuada anteriormente não foi evasiva. Sendo que, seria impossível considerar todas as abordagens possíveis, na criação dos modelos.

A aplicação de técnicas como *Random Search* ou *Grid Search*, revelam-se menos eficientes do que a utilização de algoritmos de otimização, neste caso o *PSO*, visto que não assumem uma aprendizagem iterativa e em grupo, na tentativa de identificar as melhores soluções para um determinado problema. Para além disso testam apenas um nº mínimo de situações possíveis (dada o elevado poder temporal exigido para testar todas as hipóteses possíveis).

O *PSO* iterativamente tenta melhorar a qualidade da solução atual, considerando assim uma procura inteligente das partículas, ao longo do espaço do problema, tendo em conta as ações desempenhadas por todo o grupo. O objetivo do grupo passa pela procura da solução ideal do problema, tentando assim todas as partículas convergiram para uma solução "ideal". Os motivos das partículas têm em consideração uma função objetivo. Neste problema em concreto, o objetivo das partículas passa por melhorar as métricas do problema, mais concretamente a *macro average recall* e *precision*, e para além disso, reduzir o poder computacional do modelo, isto é reduzir o nº de filtros e de neurónios, respetivamentes das camadas convolucionais e *Dense*.

Segue-se então a aplicação do *PSO* na tentativa de otimização das três arquiteturas descritas ao longo do estudo. De salientar, que irá ser otimizados os modelos, recorrendo a duas topologias de organização das partículas, ao longo do espaço, isto é, seguindo a topologia *global best* e

local best. Tentando dessa forma, analisar qual a topologia que permite a obtenção de modelos mais adequados, tendo em conta o problema em análise.

Referências:

[1] https://arxiv.org/ftp/arxiv/papers/1803/1803.10417.pdf --> The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions

[2] https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000