



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS



EMPREGO DE INTELIGÊNCIA ARTIFICIAL NA ESCOLHA AUTOMÁTICA DE ALGORITMOS E PARÂMETROS DE TÉCNICAS DE PROCESSAMENTO DE IMAGENS OBTIDAS POR DRONES PARA APLICAÇÃO NO SENSORIAMENTO REMOTO

Hércules Carlos Dos Santos Pereira

Relatório de andamento de
projeto de Iniciação Científica,
orientado pelo Dr. Valdivino
Alexandre de Santiago Júnior
e Dr. Elcio Hideiti Shiguemori.

INPE
São José dos Campos
2021

RESUMO

Veículos Aéreos Não Tripulados (VANTs), ou drones, têm recebido bastante atenção da indústria e da academia, onde estão sendo usados em aplicações diversas, tais como agricultura, transporte e logística, para a área aeroespacial, entre outras. A classificação automática de imagens obtidas por drones é importante para, por exemplo, melhorar a autonomia destes sistemas no que tange à resposta a desastres e situações de emergência em áreas de difícil acesso. O objetivo dessa pesquisa é investigar a classificação de imagens obtidas por drones utilizando Inteligência Artificial e técnicas de processamento de imagens. A primeira etapa da pesquisa se baseou em uma continuidade de um trabalho de doutorado, onde este resultou em um sistema autoadaptativo para selecionar, de forma inteligente, os melhores algoritmos de processamento digital de imagens para estimação de posição de drones por imagens. Então, fez-se uma avaliação dos seguintes algoritmos de Aprendizado de Máquina nesta pesquisa: Redes Neurais Artificiais (Rosenblatt de 1958, 1962), *AdaBoost* (Freund e Schapire, 1996), *Logistic Regression* (Hosmer e Stanley, (2000) e Árvores de Decisão(ref). Os resultados demonstraram que as Redes Neurais Artificiais de uma e duas camadas tiveram os melhores resultados se comparado as demais técnicas, atingindo mais de 95% de acerto considerando-se diversos indicadores. Na segunda etapa, o objetivo foi verificar qual combinação de extrator de características de imagens e classificador tem o melhor desempenho para classificar imagens de drones. Nesse caso, considerou-se uma classificação multiclasse com 4 classes onde também foi realizado um processo de aumento de dados (data augmentation) (Perez e Wang, 2017) para o conjunto de treinamento. Como extrator de características, foram usadas as seguintes Redes Neurais Convolucionais: *Inception v3* (SZEGEDY et al., 2015), *SqueezeNet* (Iandola et al., 2016), *VGG-16* (Simonyan e Zisserman, 2015), *VGG-19* (Simonyan e Zisserman, 2015), *Painters* (Ilenič, 2016) e *DeepLoc* (SØNDERBY, 2017). Como classificadores, foram usados *AdaBoost*(Freund e Schapire, 1996), *Random Forests* (Ho, 1965), *Logistic Regression* (Hosmer e Stanley, 2020) e Redes Neurais Artificiais (Rosenblatt de 1958, 1962). Os resultados demonstraram que a rede convolucional *VGG-16* (Simonyan e Zisserman, 2015) usada para extração de característica apresentou os melhores resultados combinando-se com o classificador neural.

Palavras-chave: Inteligência artificial, Processamento de imagens, Aprendizado de máquina, Rede neural artificial, Drones.

LISTA DE FIGURAS

Figura 1 - Cronograma de Atividades.....	9
Figura 2- Modelo de Neurônio artificial.....	13
Figura 3 - Técnica de aumento de dados	14
Figura 4 - Tabela de classificação	19
Figura 5 - código Orange Teste dos dados	19
Figura 6 – Widget de rede neural	20
Figura 7 – Widget de Logistic Regrsson	21
Figura 8 - Widget de Tree.....	21
Figura 9 - Widget de Adaboost.....	22
Figura 10 – Gráfico de Rede neural de 1 camada.....	24
Figura 11 - Gráfico de Rede neural de 2 camada.....	24
Figura 12 – Gráfico de AdaBoost	25
Figura 13 – Gráfico de Modelo Tree (Árvore de decisão).....	26
Figura 14 - Gráfico de Modelo Logistic Regression.....	27
Figura 15 - Código Orange para testa com duas entradas de imagem	29
Figura 16 - Gráfico do modelo Rede Neural Abordagem 2	31
Figura 17 - Gráfico do modelo Logistic Regression Abordagem 2	32
Figura 18 - Matriz de confusão modelo Logistic Regression	32
Figura 19 - Imagens classificadas erradas pelo modelo Logistic Regression	33
Figura 20 - Gráfico do modelo Random Forest Abordagem 2.....	34
Figura 21 - Matriz de confusão do modelo Logic Regression.....	35
Figura 22 - Imagens classificadas erradas pelo modelo Random Forest	35
Figura 23 - Matriz de confusão modelo AdaBoost.....	36
Figura 24 - Gráfico do modelo AdaBoost Abordagem 2	37

LISTA DE TABELA

Tabela 1- ETAPAS CONCLUÍDAS9

Tabela 2 - Resultado de treinamento de Rede Neural23

Tabela 3 - Resultado de treinamento de AdaBost25

Tabela 4 - Tree (Árvore de decisão)26

Tabela 5 - Resultados do Modelo Logistic Regression Abordagem 127

Tabela 6 - Resultados do Modelo Rede Neural artificial abordagem 230

Tabela 7 - Resultados do Modelo Logistic Regression abordagem 231

Tabela 8 - Resultados do Modelo Random Forest33

Tabela 9 - Resultados do Modelo AdaBoost abordagem 236

SUMÁRIO

1 INTRODUÇÃO	7
2 CRONOGRAMA DE ATIVIDADES E ETAPAS CONCLUÍDAS	8
3 MATERIAIS E MÉTODOS	12
3.1 MÉTODOS	12
3.1.2 APRENDIZAGEM DE MÁQUINA	12
3.1.3 REDE NEURAL ARTIFICIAL (RNA).....	13
3.1.5 AUMENTO DE DADOS.....	14
3.2 FERRAMENTAS COMPUTACIONAIS	14
3.2.1 AMBIENTE ORANGE	15
3.2.2 AMBIENTE COLABORATORY	15
3.2.3 SPYDER:.....	15
3.3 MATERIAIS.....	15
3.3.1 IMAGENS OBTIDAS POR DRONES.....	16
4.1 TREINO DE RNA	17
4.2 TESTES DE DIFERENTES PARÂMETROS DE RNA.....	17
4.3 EMPREGOS DE REDES NEURAS CONVOLUCIONAIS NA EXTRAÇÃO DE ATRIBUTOS.....	17
5 DESENVOLVIMENTO.....	18
5.1 ETAPA 1	18
5.1.2 RESULTADOS.....	23
5.2 ETAPA 2	28
5.2.1 RECORTE CLASSES	28
5.2.2 RESULTADOS.....	30
REFERÊNCIAS.....	39

1 INTRODUÇÃO

Os drones também conhecidos como Aeronaves Remotamente Pilotadas (ARP) ou Veículo Aéreos Não Tripulado (VANT), possuem diversas aplicações e têm sido utilizados nas mais variadas áreas. Com isso, são usados em diferentes cenários de voo com variações na cobertura do solo, terrenos, horários do dia e condições de voo, como altura e velocidade. Além disso, existem diferentes tipos de aeronaves, tais como as de asa rotativa, asa fixa e os híbridos (ROOS, LORENA e SHIGUEMORI, 2016).

Nas aplicações de sensoriamento remoto, a variação de cenários de voo dificulta no processo de automatização da classificação das imagens, sendo assim, não há um algoritmo único de processamento de imagens ou visão computacional para as diversas aplicações (Ross, Lorena e Shiguemori, 2016). Sendo assim, os parâmetros e os algoritmos variam conforme os cenários e condições de voo (Fornari, Santiago Junior e Shiguemori, 2018).

Portanto, a classificação automática de imagens obtidas por drones é importante para, por exemplo, melhorar a autonomia destes sistemas no que tange à resposta a desastres e situações de emergência em áreas de difícil acesso. O objetivo dessa pesquisa é investigar a classificação de imagens obtidas por drones utilizando Inteligência Artificial e técnicas de processamento de imagens. A primeira etapa da pesquisa se baseou em uma continuidade de um trabalho de doutorado, onde este resultou em um sistema autoadaptativo para selecionar, de forma inteligente, os melhores algoritmos de processamento digital de imagens para estimação de posição de drones por imagens. Então, fez-se uma avaliação dos seguintes algoritmos de Aprendizado de Máquina nesta pesquisa: Redes Neurais Artificiais (Rosenblatt de 1958, 1962), AdaBoost (Freund e Schapire, 1996), Logistic Regression (Hosmer e Stanley, 2000) e Árvores de Decisão (Ho, 1995).

Na segunda etapa, o objetivo foi combinar extratores de características de imagens e classificadores considerando-se imagens capturadas por drones. Nesse caso, considerou-se uma classificação multiclasse com 4 classes onde também foi

realizado um processo de aumento de dados data augmentation(Perez e Wang, 2017) para o conjunto de treinamento. Como extrator de características, foram usadas as seguintes Redes Neurais Convolucionais: *Inception v3*, *SqueezeNet*, *VGG-16*, *VGG-19*, *Painters* e *DeepLoc*. Como classificadores, foram comparadas as técnicas: *AdaBoost*, *Random Forests*, *Logistic Regression* e *Redes Neurais Artificiais*.

2 CRONOGRAMA DE ATIVIDADES E ETAPAS CONCLUÍDAS

Assim como disposto no “formulário de proposta de bolsa PIBIC”, as atividades previstas para atingir os objetivos do projeto são descritas a seguir:

1. Obter informações de diferentes sensores obtidos em voos de Drones;
2. Estudar o ambiente Orange e linguagem Python para análise dos dados. Analisar os dados;
- 3 Estudar o ambiente Orange e linguagem Python para tratamento dos dados. Tratar os dados;
4. Aplicar redes neurais artificiais (rasas ou profundas) para realizar a fusão de dados. Observação: essa atividade foi modificada para: Avaliar algoritmos de Aprendizado de Máquina para classificação automática de imagens;
5. Escrever relatório parcial;
6. Estudar bibliotecas Keras(Keras, 2021), TensorFlow (Tensorflow, 2021), Scikit Learn(Scikit-learn, 2021). Aplicar técnicas de aprendizagem de máquina;
7. Ajustar os parâmetros das técnicas;
8. Comparar desempenho das técnicas via avaliação rigorosa;
9. Submeter artigo para conferência e/ou workshop e/ou simpósio na área de Aprendizado de Máquina ou Inteligência Artificial, e elaborar relatório final de atividades.

O cronograma para desenvolvimento das atividades é demonstrado na Figura 1. O número das atividades está conforme os números mostrados acima, as colunas do Ano 1 representam os meses.

Figura 1 - Cronograma de Atividades

Atividade	Ano I											
	1	2	3	4	5	6	7	8	9	10	11	12
1	■											
2		■										
3			■									
4				■	■							
5					■							
6						■						
7							■	■				
8								■	■	■		
9											■	■

Fonte: Autoria própria (2021).

Dessa forma, esse relatório compreende o mês 1 (setembro/2020) do Ano I ao mês 11 (julho/2021) do Ano I do projeto. Considerando as atividades previstas para serem desenvolvidas, conforme a Figura 1, a Tabela 1 exibe as atividades concluídas e a concluir considerando o período referente a este relatório.

Tabela 1- ETAPAS CONCLUÍDAS E A CONCLUIR

	Atividades da Metodologia	Previsão	Realização
1	Obter informações de diferentes sensores obtidos em voos de Drones;	100%	90%
2	Estudar o ambiente Orange e linguagem Python para análise dos dados. Analisar os dados;	100%	80%
3	Estudar o ambiente Orange e linguagem Python para tratamento dos dados. Tratar os dados;	100%	80%
4	Aplicar redes neurais artificiais (rasas ou profundas) para realizar a fusão de dados. Observação: essa atividade foi modificada para: Avaliar	100%	70%

	algoritmos de Aprendizado de Máquina para classificação automática de imagens;		
5	Escrever relatório parcial;	100%	100%
6	Estudar bibliotecas Keras, Tensor Flow, Scikit Learn . Aplicar técnicas de aprendizagem de máquina;	100%	50%
7	Ajustar os parâmetros das técnicas;	100%	20%
8	Comparar desempenho das técnicas via avaliação rigorosa;	100%	40%
9	Submeter artigo para conferência e/ou workshop e/ou simpósio na área de Aprendizado de Máquina ou Inteligência Artificial, e elaborar relatório final de atividades.	50%	0%

Na Tabela 1, a coluna **Previsão** mostra a porcentagem prevista para a realização da atividade, e a coluna **Realização** mostra a porcentagem que foi realizada da atividade, considerando a que se refere esse relatório (1 de setembro de 2020 a 31 de julho de 2021).

Na atividade 1: obtenção das informações de diferentes sensores obtidos em voos de Drones, já disponíveis em repositórios do IEAv. Nesta etapa, foi realizada a análise do problema e dos dados, separando as imagens, considerando o tamanho, a cobertura do solo, a altura de voo e a resolução.

Na atividade 2, foram estudados o ambiente *Orange* (Universidade Ljubljana, 2021) e a linguagem *Python* (Python, 2021), *Spyder* (Spyder IDE, 2020), além do

processamento em nuvem *Colaboratory* (Google Cloud, 2021). O objetivo foi a familiarização com as ferramentas de aprendizagem de máquina e análise dos dados (Universidade Ljubljana, 2021), (Python, 2020). Nestes ambientes foram exploradas as ferramentas para visualização dos dados.

Na atividade 3, foi realizado o tratamento dos dados. Os dados foram separados com uso do ambiente Orange. Além disso, com uso da linguagem *Python*, retirou os dados faltantes e removeu dados julgados menos relevantes.

A atividade 4 devido à falta de dados de voo relacionados com as imagens obtidas pelos Drones, aprofundou-se nas técnicas de aprendizado de máquina utilizadas por (Fornari, 2020). foi: Redes Neurais Artificiais, *AdaBoost*, *Logistic Regression*, Árvores de Decisão, *Random Forests*. Além das técnicas para extração de atributos: Redes Neurais Convolucionais, *Inception v3*, *SqueezeNet*, *VGG-16*, *VGG-19*, *Painters* e *DeepLoc*.

A atividade 5 consistiu na elaboração deste relatório.

A atividade 6 consistiu no estudo das bibliotecas *Keras* (keras, 2021), *Tensor Flow* (Tensor Flow, 2021), *Scikit Learn* (Scikit-Learn, 2021). Nesta etapa No entanto, uma análise mais aprofundada e criteriosa será feita no próximo ano de renovação da Iniciação Científica;

Na atividade 7 foram testados os hiper-parâmetros dos modelos de Aprendizagem de Máquina: Rede neural artificial, *Adaboost*, *Random Forest*, *Tree* e *Regression logistic*. Estes ajustes foram realizados com uso do Orange, mas também serão realizadas buscas sistemáticas para os parâmetros ideais para o problema e dados considerados;

Para a atividade 8, a avaliação do desempenho das técnicas foi realizada com uso do ambiente Orange. A avaliação mais rigorosa também será realizada no próximo ano.

Na atividade 9 é proposto submeter um artigo científico para o GEOINFO 2021. O Simpósio Brasileiro de Geoinformática (GeoInfo, 2021) é uma conferência que acontece anualmente com o intuito de explorar pesquisas, desenvolvimento e aplicações inovadoras para ciência da informação geográfico e áreas aproximadas (GeoInfo, 2021). O artigo ainda precisa ser elaborado.

Portanto, pode-se dizer as atividades do projeto foram desenvolvidas de forma bastante satisfatória nesse período. Mas, principalmente com a redefinição da atividade 4, a qual demandou muito tempo, é necessário renovar a bolsa por mais 12 meses para que a pesquisa alcance seu objetivo completamente.

Os materiais e as metodologias das atividades realizadas são apresentados a seguir.

3 MATERIAIS E MÉTODOS

O desenvolvimento do projeto foi dividido em 2 etapas. A primeira consistiu em utilizar técnicas computacionais considerando-se a metodologia proposta por Fornari, (2020). A segunda consistiu no uso de técnicas de aprendizagem de máquina. A seguir é apresentada uma lista dos métodos e matérias empregados nesta pesquisa.

3.1 Métodos

Os métodos empregados na pesquisa aprendizagem de máquina, redes neurais artificiais, redes pré-treinadas e Técnica de Aumento de Dados.

3.1.2 Aprendizagem de máquina

Machine learning ou aprendizagem de máquina é uma ramificação da área de inteligência artificial. O método também é usado na análise de dados e se baseia na ideia de que um sistema ou máquina tem capacidade de aprender através de dados, ou seja, o modelo se adapta e aprende a analisar padrões. Alguns dos tipos de

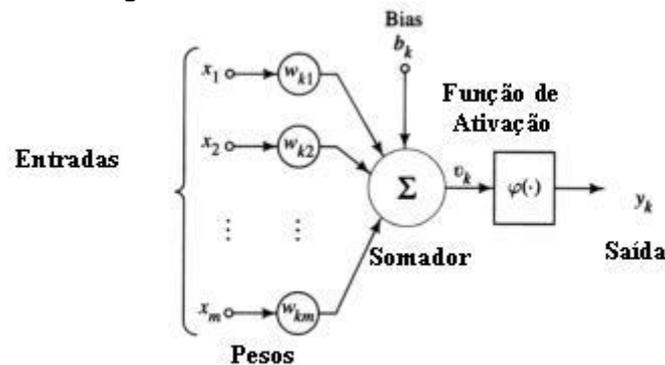
aprendizado de máquina são: o aprendizado supervisionado, não supervisionado, semi-supervisionado e por reforço segundo Hurwitz e Kirsch,(2018).

O método utilizado nos experimentos deste trabalho foi o aprendizado supervisionado, nessa abordagem se tem um conjunto de dados apresentados como entrada, também é apresentada a saída (Hurwitz e Kirsch , 2018).

3.1.3 Rede Neural Artificial (RNA)

As RNA se baseiam em neurônios biológicos. Uma rede neural artificial (Rosenblatt de 1958, 1962) é composta por vários neurônios artificiais, que são modelos matemáticos. Eles possuem entradas, pesos, processamento e função de ativação que será enviada a saída. Como entrada são as características dos problemas. Ela é multiplicada pelo seu respectivo peso e o somador realiza o processamento, que é a soma das multiplicações das entradas vezes o peso. A função de ativação é responsável por ativar ou inibir a intensidade do neurônio que serve como o resultado na saída. Na Figura 2 é ilustrado um neurônio artificial.

Figura 2- Modelo de Neurônio artificial



Fonte: Saulo Popov Zambiasi (2008).

As redes neurais artificiais são compostas por vários neurônios e camadas. As redes neurais profunda possuem diversas camadas e diversos neurônios. É importante definir a topologia mais adequada para um problema e os dados, com isso deve-se

utilizar hiper-parâmetros adequados como o número de neurônios, camadas e funções de ativação (Zambiasi, 2008).

3.1.5 Aumento de Dados

De modo geral, o treinamento eficaz de redes neurais artificiais requer muitos dados. No caso de poucos dados, as técnicas de aumento de dados ou Data Augmentation são indicadas (Perez e Wang, 2017). A Figura 3 ilustra o resultado da aplicação das técnicas de aumento de dados em imagens. O método usado em imagem consiste em rotacionar, transformar, aplicar ruídos, ampliar imagem, escurecer e destorcer (Perez e Wang, 2017). Esse método aumenta consideravelmente um conjunto de dados.

Figura 3 - Técnica de aumento de dados



Fonte: Fonte: Autoria própria (2021).

3.2 Ferramentas computacionais

Nesta seção são apresentadas as ferramentas que foram usadas na pesquisa, os softwares e os ambientes de desenvolvimento.

3.2.1 Ambiente Orange

O Orange é um software de código aberto desenvolvido em Python (Python, 2021) usado também em mineração de dados, *machine learning* e análise de dados. Nele não é necessária a programação do código, já que conta com uma diversidade de ferramentas, tanto de visualização, quanto de processamento de dados. No ambiente podem ser utilizadas diferentes técnicas de aprendizado de máquina. Com o software é possível treinar, testar e analisar o desempenho de diferentes modelos.

3.2.2 Ambiente Colaboratory

O Google *Colaboratory* é um ambiente que funciona na nuvem baseado no *Jupyter Notebook* (Google Cloud, 2021), focado em estudo e trabalho com ciências de dados, inteligência artificial utiliza a linguagem *Python* (Python, 2021). O software necessita como recursos o navegador e o acesso à internet. Os blocos de código são executados em máquinas virtuais que são reservados por conta, possui recurso de acelerador de hardware, GPU para melhorar a aceleração de cálculo gráficos, TPU que é um circuito integrado desenvolvida pela google para acelerar o aprendizado de máquina.

3.2.3 Spyder:

O *Spyder* (Spyder IDE, 2020) é uma IDE de código aberto criada em *Python* e que suporta códigos na própria linguagem *Python* (Python, 2021). possui vários tipos de ferramentas e recursos de manipulação de visualização de dados voltados para aplicações científicas (Spyder IDE, 2020). O *Spyder* (Spyder, 2021) foi utilizado no recorte das imagens para gerar as classes que servem como entradas para os modelos de aprendizagem de máquina, após o recorte as imagens são dispostas em uma pasta de seus respectivos rótulos.

3.3 Materiais

3.3.1 Imagens obtidas por drone.

Um conjunto com vinte e cinco imagens de áreas urbanas obtidas por uma Aeronave Remotamente Pilotada foi utilizado. As imagens possuem 5 cm de resolução, quatro mil pixels de largura e três mil de comprimento. Esse conjunto foi usado para extrair as classes de treinamento que se baseou no trabalho de Marielcio (2020). As classes que foram extraídas com técnicas de processamento de imagens, separando-as por classes de casa, carro, árvore e prédio.

4 Experimentos

O primeiro experimento tem o intuito de buscar os classificadores com melhor desempenho. Foram analisados os dados de sensores obtidos em voo de drones. Os arquivos resultantes são em formato txt. As colunas possuem informação de variância, energia, contraste de imagem, desvio padrão, contraste máximo, altitude de voo do drone e os diferentes algoritmos de classificação. Estes processos foram baseados no trabalho de Fornari (2020).

No ambiente Orange foi realizado o pré-processamento, com a seleção das colunas que servem como uma variável alvo. O próximo seguinte consiste em ajustar os diferentes modelos de aprendizagem de máquina, nesse experimento foram testados e comparados os modelos: rede neural artificial com uma e duas camadas, *Logistic Regression*, *Tree* Árvore de decisão e *AdaBoost*. Após os testes foram gerados uma tabela para cada um dos modelos contendo os resultados gerados pelo treinamento a fim de comparação.

O experimento dois foi realizado com imagens obtidas por voo de drone a fim de determinar os classificadores de imagem que apresentam melhores desempenhos na classificação das imagens. As imagens dispostas em formato jpg com dimensão de 3000 x 4000 pixels obtidas uma câmera com resolução de 5 cm.

Algumas imagens foram analisadas a fim de determinar quais classes seriam extraídas após a análise: árvore, casa, carro e prédio. O segundo passo foi realizar o pré-processamento no caso a extração das classes por meio de recorte foi usado a linguagem Python (Python, 2021) e a biblioteca Opencv2 (OpenCv, 2021) onde gerou 292 imagens.

O conjunto foi dividido em treinamento e teste e ambos os conjuntos ficaram com 146 imagens, considerados pequenos para serem usados no treinamento. Para

aumentar o conjunto foi proposto usar técnicas de aumento de dados que consistem em usar rotação em 90 e 180 graus, espelhamento da imagem, acrescentar ruídos e aumentado contraste as combinações dessas técnicas resultou em 3005 imagens. No *Orange* foram criadas duas entradas uma para o conjunto de treinamento e outra para o teste. Para extrair as características das imagens foram utilizadas redes neurais convolucionais a *Inceptionv3*, *SqueezeNet*, *VGG-16*, *VGG-19* e *DeepLoc*, modelos usados para o treinamento do modelo foram a *Logistic Regression*, *Ranom Forest* e o *AdaBoost* onde cada técnica teve seu parâmetro variado, e após o treinamento foi criado uma tabela para cada um dos modelos com os resultados gerados pelo “*Test And Score*” no *Orange*. De cada tabela com os resultados foi calculada a média. Em seguida, gerou-se os gráficos para representar o desempenho dos modelos, também foram anotados os 5 melhores desempenho de cada técnica.

4.1 Treino de RNA

Para o treinamento da rede neural artificial foi usado o ambiente *Orange*. Foram testadas combinações de diferentes parâmetros: funções de ativação, número neurônios, função de ativação, número de épocas de treinamento e otimizadores.

4.2 Testes de diferentes parâmetros de RNA

O experimento consiste em testar a combinação dos valores, os parâmetros testados no modelo foram, o número de neurônio variando 20 a 250 para rede com uma camada e para duas se usou os valores 20, 20 a 80, 100, foram testados 4 tipos de função de ativação, 3 otimizadores de código e o número de interação entre 1000, 1500, 2000 e 3000.

4.3 Empregos de redes neurais convolucionais na extração de atributos

A extração das características foi realizada através um “widget” de extração de características de imagem disposto no *Orange* sendo realizados pelos modelos de redes convolucionais, foi testado seis tipo de modelo, o *Inception v3*, *SqueezeNet(local)*, *VGG-16*, *VGG-19*, *Painters* e *DeepLoc*.

Os modelos de reconhecimento de imagem *Inception v3* o modelo desenvolvido pela *Google* para a incorporação das imagens, é ativação na penúltima camada onde contem a representação vetorial da imagem, a técnica foi treinada com o conjunto *imageNet*(mageNet, 2021), (Universidade de Ljubljana, 2020).

O *SqueezeNet* é uma rede profunda com precisão equivalente ao modelo *AlexNet* só que com 50 vezes menos parâmetros, para a incorporação das imagens é usado ativação na última camada, O modelo é considerado pequeno, mas com um alto desempenho, foi treinado com data set *imageNet* (ImageNet, 2021).

Os modelos *VGG-16* e a *VGG-19* são compostas por 16 e 19 camadas em ambos os modelos foram treinados com banco de imagens *imageNet* (ImageNet, 2021). Para incorporador é usado a ativação na penúltima camada (Universidade Ljubljana, 2021).

Já o modelo *Painters* é os modelos usados para detectar pinturas por meio de imagens, o modelo foi treinado com 79.433 imagens de 1.684 pintores diferente. Esse modelo foi proposto pelo *Geometry Grup* da universidade de *Oxford*.

O modelo *DeepLoc* é uma rede *convolucional* que foi treinada com um conjunto de 21.882 imagens de células únicas geradas de forma manual. O incorporador é ativado na penúltima camada do modelo (Universidade Ljubljana, 2021).

5 DESENVOLVIMENTO

Nesta seção são apresentados os processos de desenvolvimento usados na pesquisa, os materiais, os métodos, as análises e os resultados obtidos.

5.1 Etapa 1

Na primeira etapa foi realizado um teste com os modelos de Aprendizagem de máquina, rede neural, *Logistic Regression*, *Tree* e *Adadboost*. em um conjunto com 2163 dados obtidos por voo de drone por Fornari,(2020), esta abordagem tem o intuito de comparar quais as técnicas de aprendizagem de máquina possuem os melhores resultados na classificação de imagem.

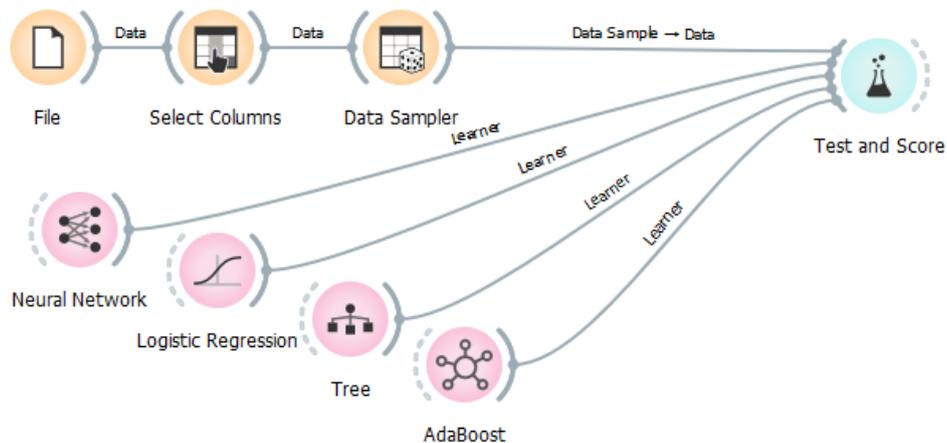
Figura 4 - Tabela de classificação

Id	Entropy H	Difference	Dissimilarity	Energy	Homogeneity	Contrast	Average	Std Dev	Entropy	Max Contr	Altitude	Pitch	Roll	Algorithm Class
1	-1.2255	0.0090	0.0875	8.0122	0.9846	0.0167	107.87	59.6097	7.6877	0	31.90	0.00	0.00	307
2	-1.0827	0.0124	0.1073	7.0981	0.9782	0.0239	110.64	60.2336	7.7087	0	31.90	0.00	0.00	307
3	-1.1644	0.0128	0.1040	8.3687	0.9785	0.0236	107.48	60.1447	7.6871	0	31.90	0.00	0.00	307
4	-1.2918	0.0122	0.0922	9.0399	0.9812	0.0207	107.84	61.4786	7.7014	0	31.90	0.00	0.00	307
5	-1.6035	0.0091	0.0693	12.1377	0.9873	0.0139	110.92	62.3206	7.6480	0	31.90	0.00	0.00	307
6	-1.9005	0.0052	0.0484	14.3127	0.9930	0.0075	111.22	59.0765	7.5747	0	31.90	0.00	-0.10	1906
7	-2.2414	0.0024	0.0343	17.0754	0.9966	0.0036	107.24	52.5158	7.4726	0	31.90	0.00	0.00	507
8	-2.4146	0.0017	0.0311	18.7204	0.9975	0.0026	103.57	45.8734	7.3495	1	31.90	0.00	-0.10	507
9	-2.4138	0.0015	0.0327	18.0121	0.9975	0.0026	102.27	43.0909	7.3151	1	31.90	0.00	-0.10	507
10	-2.3099	0.0018	0.0355	17.0920	0.9970	0.0031	103.40	44.7380	7.4134	1	31.90	0.00	0.00	507

Fonte: Gabriel Fornari (2020).

Os dados usados na primeira abordagem foram obtidos por sensores em voo de drone (Fornari, 2020), são ilustrados na Figura 4, as colunas possuem informação de variância, energia, contraste de imagem, desvio padrão, contraste máximo, altitude de voo do drone e os diferentes algoritmos de classificação Fornari (2020).

Figura 5 - código Orange Teste dos dados



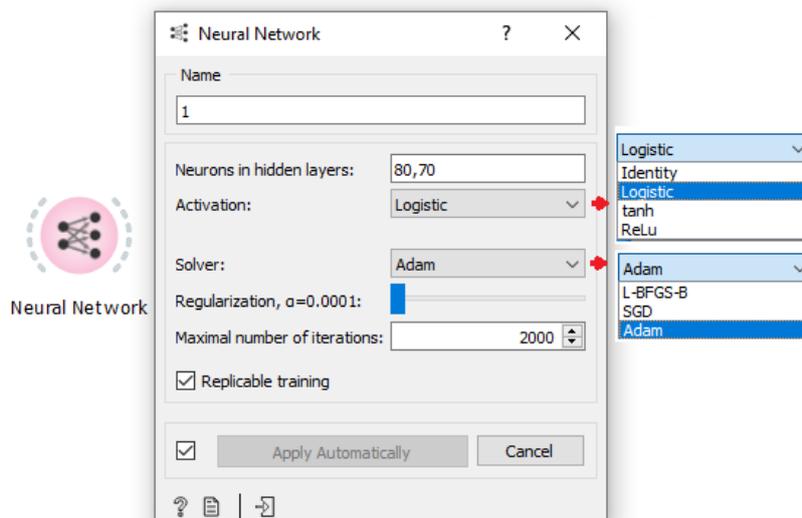
Fonte: Universidade de Ljubljana (2021).

Os widgets usados no espaço *Orange* se conectam por meio de linha desta forma é possível conectar *widgets* já dispostos no ambiente, O *widget* “file” realiza o carregamento dos dados no formato txt. no *widget* “select columns” é feito a separação manual de quais colunas serão utilizadas. O “data sample” realiza a

separação do conjunto de dados em treinamento e teste, o “*Teste and Score*” demonstra a pontuação da técnica vários. Esses passos são ilustrados na Figura 5.

5.1.1 Testes com os modelos ML

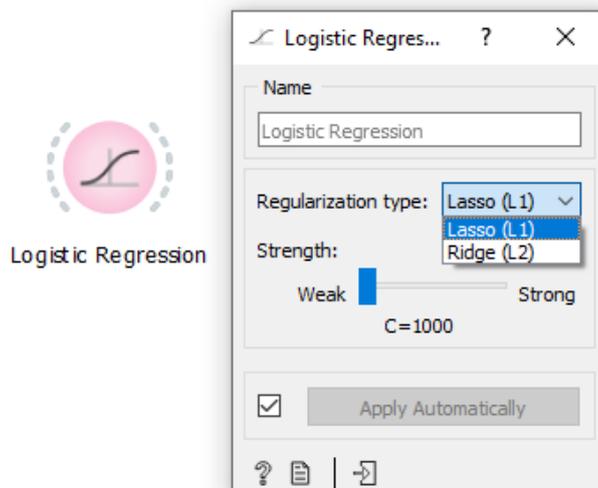
Figura 6 – Widget de rede neural



Fonte: Universidade de Ljubljana (2021).

Os parâmetros testados no modelo de rede neural foi o número de neurônio, a função de ativação, otimizador de código e o número de interação. A figura 6 ilustra o modelo no ambiente Orange e os parâmetros testados, O número de neurônio para rede de uma camada de 20 a 250, e para duas camadas os valores de 20, 20 a 80, 100, as 4 funções de ativações *Identity*, *Logistic tanh* e *Relu*, os 3 otimizador de código *L-BFGS-B*, *SDG* e *Adam* e o número de interação ficou em 1000, 1500, 2000 e 3000.

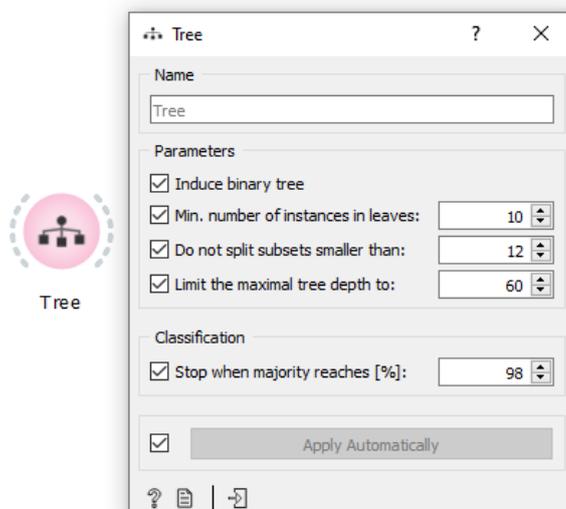
Figura 7 – Widget de Logistic Regrssion



Fonte: Universidade de Ljubljana (2021).

Os parâmetros usados no experimento com a *Logistic regression* é ilustrado na Figura 7 os respectivos parâmetros que foram usados no experimento, os dois tipos de regularizador o *Lasso(L1)* e *Ridge(L2)*, a força de custo com os valores de 0.05, 0.1 a 0.9, e de 1 a 1000.

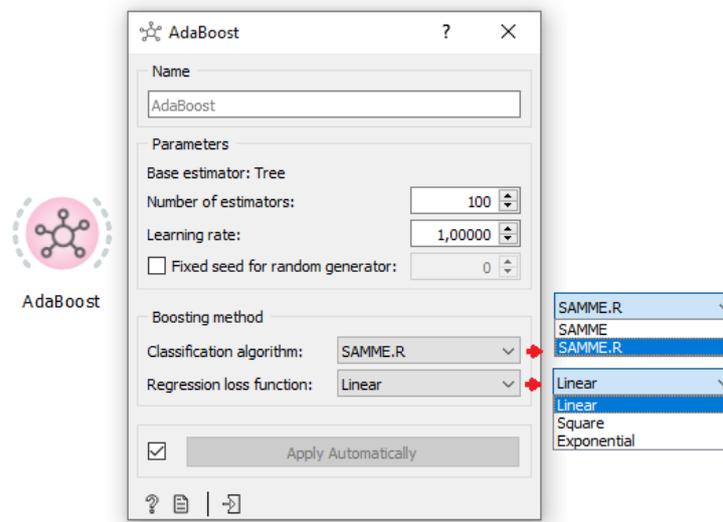
Figura 8 - Widget de Tree



Fonte: Universidade de Ljubljana (2021).

Para o modelo *Tree* árvore de decisão os parâmetros usados foram número mínimo de interação em folhas de 2, 3, 5, 6 e 10, e o número mínimo por interação de 3, 6, 10 e 12, o número máximo de divisão de 20 a 60. A figura 8 ilustra os parâmetros contínuos na técnica.

Figura 9 - Widget de Adaboost



Fonte: Universidade de Ljubljana (2021).

Para a técnica *AdaBoost* foram testados os parâmetros do número de estimadores entre 20 a 50 e 100, A taxa de aprendizado em 1 os dois algoritmos de classificação *SAMME* e *SAMME.R*, as funções de perda de regressão *Linear*, *Square*, *Exponencial*, A Figura 9 ilustra os parâmetros contidos no modelo.

5.1.2 Resultados

Para demonstrar os resultados foram gerados gráficos através da média dos valores de AUC, CA, F1, *Precision* e *Recall* com a média desses valores é possível criar um gráfico para demonstrar os resultados nos experimentos. Também foi anotado os 5 melhores resultados gerados por cada modelo.

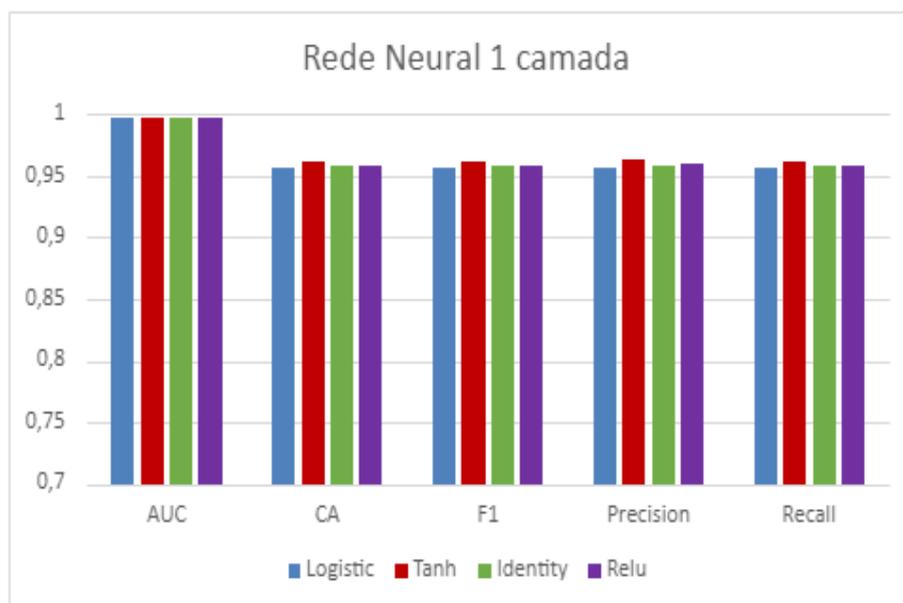
Tabela 2 - Resultado de treinamento de Rede Neural

Modelo: Rede Neural								
Neuron	Action	solver:	MxNº interattions	AUC	CA	F1	Precision	Recall
70 80	tanh	Adam	1000	0.998	0.965	0.965	0.966	0.965
80 70	tanh	Adam	1000	0.998	0.965	0.965	0.965	0.965
80 70	tanh	Adam	2000	0.998	0.965	0.965	0.965	0.965
80	tanh	Adam	1500	0.998	0.965	0.964	0.965	0.965
80	tanh	Adam	2000	0.998	0.965	0.964	0.965	0.965

Fonte: Produção do autor (2020).

O melhor resultado obtido do modelo rede neural de uma camada e composto por 80 neurônios, com uma função de ativação do tipo tangente hiperbólica, um otimizador de código *Adam* e com 2000 interações isso é demonstrado na Tabela 2 na quarta linha. Já para a rede neural de duas camadas a pontuação mais alta é obtida da combinação com 70 e 80 neurônios, com a função de ativação tangente hiperbólica, o otimizador de código *Adam*, e com 1000 interações como demonstrado na Tabela 2 na primeira linha.

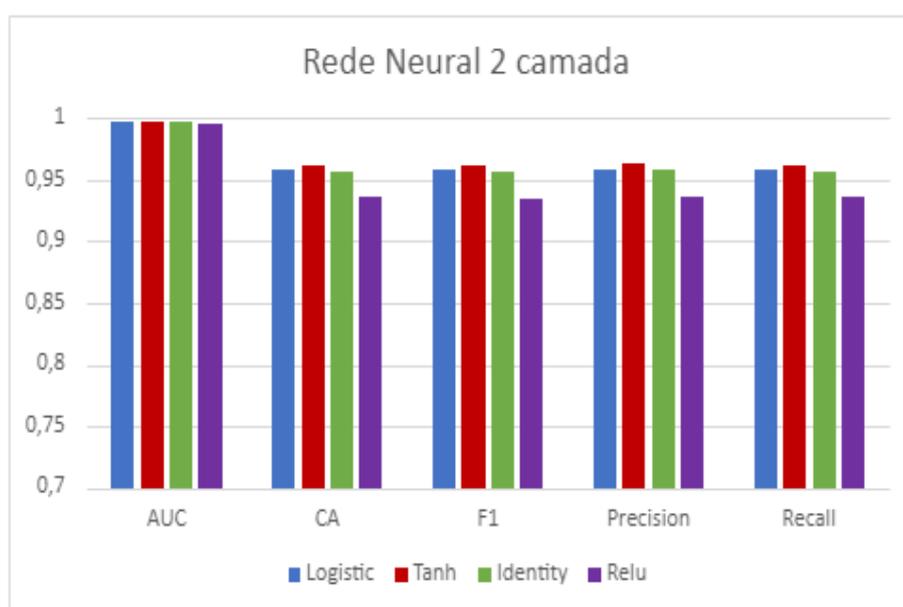
Figura 10 – Gráfico de Rede neural de 1 camada



Fonte: Autoria própria (2021).

No experimento com a rede neural de uma camada a função de ativação que demonstrou a maior pontuação foi tangente Hiperbólica como é ilustrado na Figura 10 é ilustrado os resultados nos valores de AUC, CA, F1, *Precision* e *Recall*. É importante ressaltar que nesse experimento o otimizador código ficou fixo *Adam*, já que após testar com os outros otimizadores os valores foram muito baixos em comparação ao Adam.

Figura 11 - Gráfico de Rede neural de 2 camada



Fonte: Autoria própria (2021).

O teste com a rede neural com duas camadas se comparou as funções de ativações, assim como nos testes com a rede neural de uma camada a função de ativação *Tanh* tangente hiperbólica foi a que gerou os melhores resultados em *CA*, *F1*, *Precision* e *recall* como é ilustrado na Figura 11.

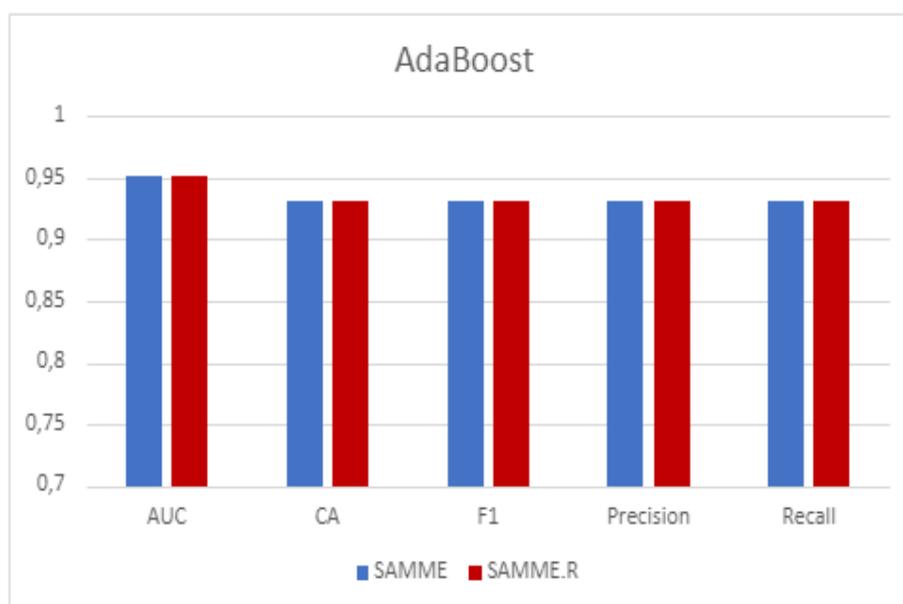
Tabela 3 - Resultado de treinamento de AdaBoost

Modelo: AdaBoost								
ClasAlg	funct/regress	Nº Estimators	Lear_rate	AUC	CA	F1	Precision	Recall
SAMME	Linear	20	1	0.952	0.931	0.931	0.931	0.931
SAMME	Square	20	1	0.952	0.931	0.931	0.931	0.931
SAMME	Exponencial	20	1	0.952	0.931	0.931	0.931	0.931
SAMME.R	Linear	20	1	0.952	0.931	0.931	0.931	0.931
SAMME.R	Square	20	1	0.952	0.931	0.931	0.931	0.931

Fonte: Produção do autor (2021).

Os valores gerados pelo modelo *AdaBoost* não foram tão altos se comparados com os demais modelos como pode ser visto na Tabela 3, tanto o classificador *SAMME* e *SAMME.R* obtiveram os mesmos valores de pontuação com diferentes funções de regressão e estimadores.

Figura 12 – Gráfico de AdaBoost



Fonte: Autoria própria (2021).

Na comparação dos algoritmos de classificação *SAMME* e *SAMME.R*, é tanto na Tabela 3 quanto na Figura 12 onde apresenta que ambos obtiveram exatamente a mesma pontuação.

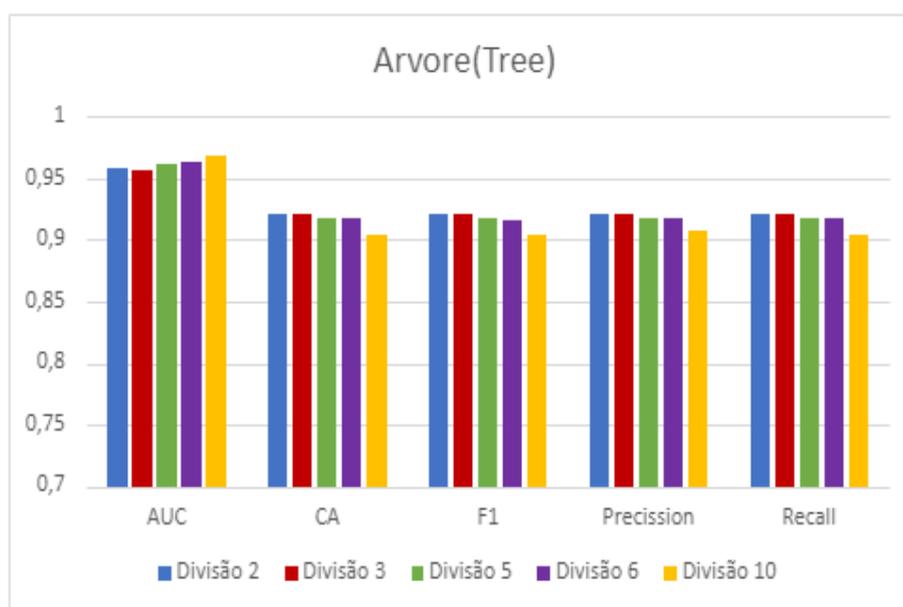
Tabela 4 - Tree (Árvore de decisão)

Modelo: Tree (Árvore)							
NMin_Interaction	DivTree	limProfundade	AUC	CA	F1	Precision	Recall
2	6	20	0.951	0.922	0.921	0.922	0.922
2	6	30	0.951	0.922	0.921	0.922	0.922
2	6	50	0.951	0.922	0.921	0.922	0.922
3	3	20	0.953	0.922	0.921	0.921	0.922

Fonte: Produção do autor (2021).

O experimento feito com a *Tree* Árvore de decisão obteve valores próximos como é disposto na Tabela 4, como é disposto na linha 1, 2 e 3 os valores gerados pelos testes onde vemos que a árvore com 2 divisões teve resultados iguais mesmo variando o limite de profundidades.

Figura 13 – Gráfico de Modelo Tree (Árvore de decisão)



Fonte: Autoria própria (2021).

Para comparar os resultados do modelo foi analisado o número de divisão de árvore como é ilustrado na Figura 13, as divisões com os valores 2 e 3 demonstraram um bom resultado se comparado com as divisões 5, 6, e 10.

Tabela 5 - Resultados do Modelo Logistic Regression Abordagem 1

Modelo: Logistic Regression						
Reguralization	Strength	AUC	CA	F1	Precision	Recall
Lasso(L1)	800	0.987	0.891	0.887	0.890	0.891
Lasso(L1)	1000	0.987	0.890	0.887	0.890	0.890
Lasso(L1)	600	0.987	0.885	0.882	0.885	0.885
Lasso(L1)	400	0.985	0.882	0.879	0.882	0.882
Ridge(L2)	0.500	0.959	0.801	0.792	0.791	0.801

Fonte: Produção do autor (2021).

O modelo Logistic Regressão retem os valores mais baixos comparado aos demais modelos isso é demonstrado na tabela 5, chegando a 79% e no máximo 89% de acerto.

Figura 14 - Gráfico de Modelo Logistic Regression



Fonte: Autoria própria (2021).

Para demonstra o resultado da técnica foi comparado os regularização *Lasso(L1)* e *Ridge(L2)* que é demonstrado na Figura 14 é possível notar que *Ridge(L2)* tem resultados melhores se comparado ao regularizador *Lasso(L1)*.

Após os experimentos foi possível notar quais técnicas de aprendizagem de máquina demonstraram os melhores resultados, sendo a rede neural que teve a

pontuação mais alta em relação aos demais modelos tanto de AUC, CA, F1, *Precision* e *Recall* para essa abordagem.

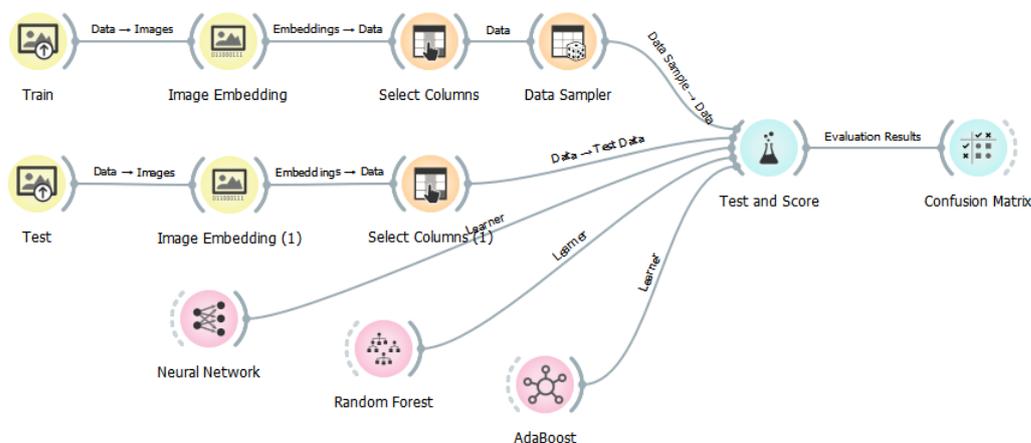
A rede neural de duas camadas com os parâmetros composto por 70, 80 neurônios com uma função de ativação do tipo *Tanh* tangente hiperbólica e com o um número de 1000 interações teve o desempenho melhor se comparada as demais funções junto ao otimizador de código Adam gerou 99% de acurácia e 96% de precisão. E que uma rede neural com uma camada composta por 80 neurônio tem desempenho próximos com 1500 interações com AUC 99% e CA, F1, *Precision* e *Recall* com 0.96%. Já o *AdaBoost* foi o que teve o segundo melhor desempenho no experimento com o classificador de algorítmico SAMME, a função de regressão Linear, e o número de estimadores em 20, os dois algoritmos de classificação tiveram o mesmo desempenho tanto em AUC com 95% e CA, F1, *Precision* e *Recall* se manteve em 93%. Já os modelos *Tree* e *logistic regression* nesse experimento tiveram os valores baixos se comparando a RNA e o *AdaBoost* tanto que a *tree* atingido AUC 95%, CA, F1, *Precision* e *Recall* com 92% e a *Logistic Regression* AUC 98%, CA 89%, F1 88%, *Precision* e *Recall* com 89% tendo o menor desempenho comparado com os demais modelos.

5.2 Etapa 2

5.2.1 Recorte classes

Foi analisado um conjunto de imagem de voo de drone sobre áreas urbanas Marielcio, (2020) para determinar quais classes seriam necessárias para o experimento, após a análise foram selecionadas as classes de árvore, casa, carro e prédio. As classes foram extraídas do conjunto por meio de um código usando linguagem Python (Python, 2021) e a biblioteca *Opencv* (OpenCv, 2021), os recortes foram dispostos em pastas separadas com os nomes denotados em suas respectivas pastas. Após o processo foi gerado um conjunto 292 imagens que foi separado em dois conjuntos de 147 para treino e 147 para teste, se aplicou as técnicas de aumento de dados que gerou 3005 imagens os dados de teste não foram alterados.

Figura 15 - Código Orange para testa com duas entradas de imagem



Fonte:

Universidade de Ljubljana (2021).

O código visual usado no experimento, para as entradas dos dados no ambiente Orange foi usado o widget “*import image*” nomeado de Train e Test é ilustrado na Figura 15 são usados para a entrada dos dados de treinamento e teste. “*Image Embedding*” é o widget responsável por extrair as características das imagens usando redes convolucionais, o “*Select columns*” serve para selecionar as colunas que serão utilizados. no “*data sampler*”. Depois os dados são passados para o “*test and score*” junto com a entrada do algoritmo de classificação (Universidade de Ljubljana, 2021) e como saída é obtido os resultados gerados na classificação. A matriz de confusão serve para visualizar as classificações que foram classificadas corretamente e as que não foram e por quais classes foram confundidas.

5.2.2 Resultados

Nesta seção será mostrado a abordagem 2, demonstrando todo o processo para a execução do experimento.

Tabela 6 - Resultados do Modelo Rede Neural artificial abordagem 2

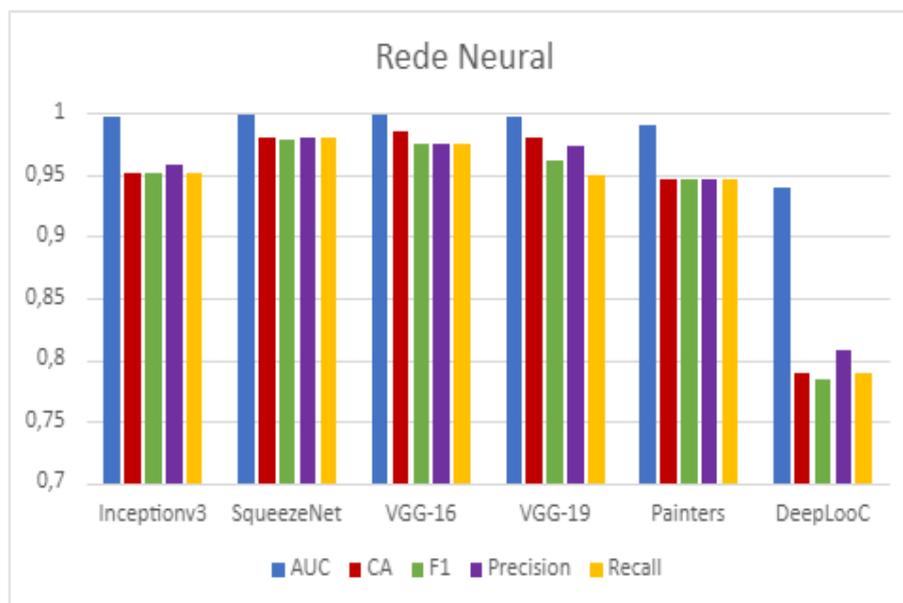
Rede Neural Artificial									
Incorporador	Action	solver	Neuron	interattions	AUC	CA	F1	Precision	Recall
VGG-16	tanh	Adam	70	2000	1	1	1	1	1
VGG-16	Relu	Adam	70	2000	1	1	1	1	1
SqueezeNet	Logistic	Adam	60	2000	1	0.993	0.993	0.993	0.993
SqueezeNet	identity	Adam	30	2000	0.999	0.993	0.993	0.993	0.993
SqueezeNet	tanh	Adam	30	2000	0.999	0.993	0.993	0.993	0.993
SqueezeNet	Relu	L-BFGS-B	30	2000	0.999	0.993	0.993	0.993	0.993

Fonte: Produção do autor (2021).

Assim como na primeira abordagem o modelo de rede neural foi testado os números de neurônio que varia entre 30, 60, 70 e 80, a função de ativação *Identity*, *logistic*, *tanh* e *Relu* os otimizadores de código L-BFGS-B, SGD, Adam o número máximo de interação ficou mantido em 2000.

Na tabela 6 é possível notar que o modelo demonstrou um desempenho de 100% na classificação das imagens sendo gerado com 70 neurônios, o incorporador VGG-16, a função de ativação *tanh*, otimizado de código adam e com 2000 interações máximas.

Figura 16 - Gráfico do modelo Rede Neural Abordagem 2



Fonte: Autoria própria (2021).

Com os resultados gerados no modelo de rede neural foi gerado um gráfico que é demonstrado na figura 16 que é resultante da medias dos resultados obtidos no treinamento do modelo, onde é possível notar que nesse experimento que o extrator de características *SqueezeNet* demonstro o melhor desempenho de AUC, CA, F1, *Precision* e *Recall* seguido pela *VGG-16*, a *VGG19*, a *inceptionV3*, *Painters* e por fim o *DeepLoc* com o rendimento mais baixo em relação aos demais incorporadores.

Tabela 7 - Resultados do Modelo Logistic Regression abordagem 2

Logistic Regression							
Incorporador	Reguralization	Strength	AUC	CA	F1	Precision	Recall
SqueezeNet	Ridge(L2)	200	1	0.993	0.993	0.993	0.993
SqueezeNet	Ridge(L2)	500	1	0.993	0.993	0.993	0.993
SqueezeNet	Ridge(L2)	1000	1	0.993	0.993	0.993	0.993
VGG-16	Ridge(L2)	0.500	1	0.993	0.993	0.993	0.993
VGG-16	Ridge(L2)	1	1	0.993	0.993	0.993	0.993

Fonte: Produção do autor (2021).

Para o modelo *Logisitic Regression* foi testado os parâmetros do regularizador em *Ridge(L2)* e *Lasso(L1)* a variação do *strength*, o tipo de regularizador em 0.500, 1, 200 e 500. Na tabela 7 são mostrados os resultados obtidos após o treinamento da

técnica *Logistic Regression* o AUC se manteve em 1 os demais resultados se mantiveram em 99%.

Figura 17 - Gráfico do modelo Logistic Regression Abordagem 2



Fonte: Autoria própria (2021).

O extrator de característica do tipo VGG-16 Demonstrou o melhor resultado nos modelos em AUC, CA, F1, *Precision* e *Recall* como é demonstrado na Figura 17. Seguido pelo *SqueezeNet* com segundo melhor resultado e com os valores mais baixos o *DeepLooc*.

Figura 18 - Matriz de confusão modelo Logistic Regression

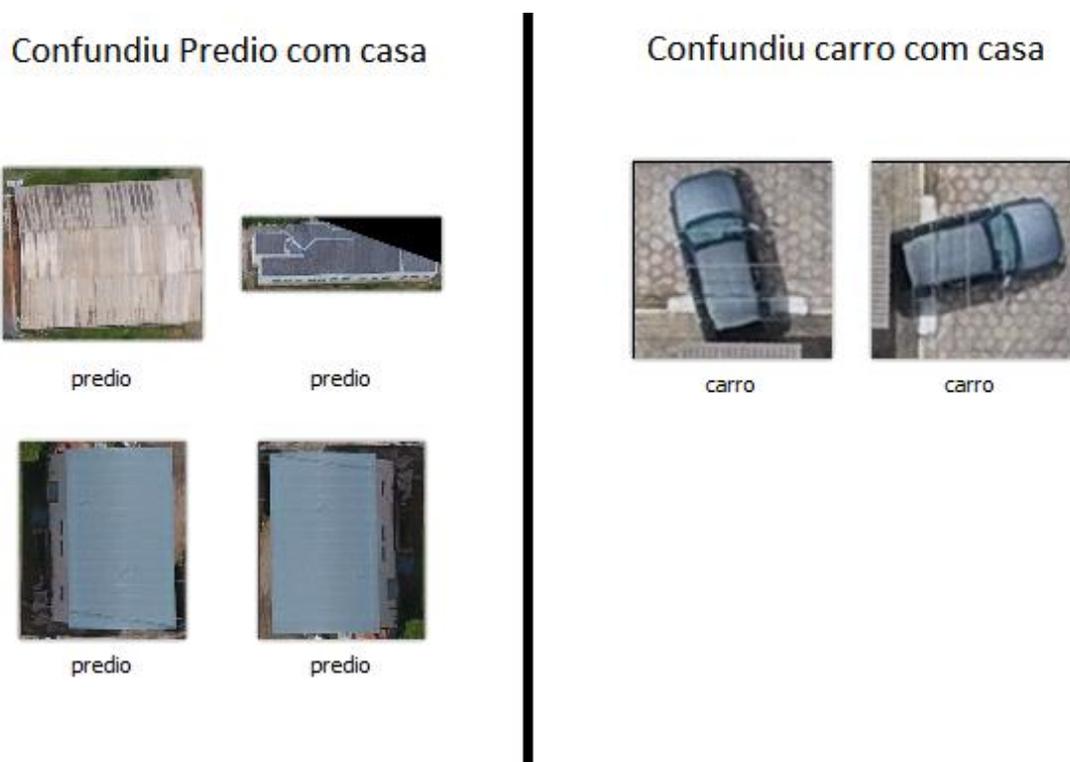
		Predicted				Σ
		arvore	carro	casa	predio	
Actual	arvore	100.0 %	0.0 %	0.0 %	0.0 %	41
	carro	0.0 %	100.0 %	3.6 %	0.0 %	40
	casa	0.0 %	0.0 %	72.7 %	0.0 %	40
	predio	0.0 %	0.0 %	23.6 %	100.0 %	26
Σ		41	38	55	13	147

Fonte: Universidade de Ljubljana (2021).

Uma matriz de confusão foi gerada com base nos valores resultante no treinamento do modelo Logistic Regression demonstrado na Figura 18, o melhor

resultado obtido pela técnica foi disposto na matriz. Na matriz é possível notar que a técnicas não teve dificuldade para classificar as classes arvore e carro, o modelo errou 3.6% carro e 23.6% de prédio como casa.

Figura 19 - Imagens classificadas erradas pelo modelo Logistic Regression



Fonte: Autoria própria (2021).

O modelo classificou de forma errada algumas imagens como é ilustrado na Figura 19 onde foi classificada a classe prédio como casa e o carro que foi confundido como casa.

Tabela 8 - Resultados do Modelo Random Forest

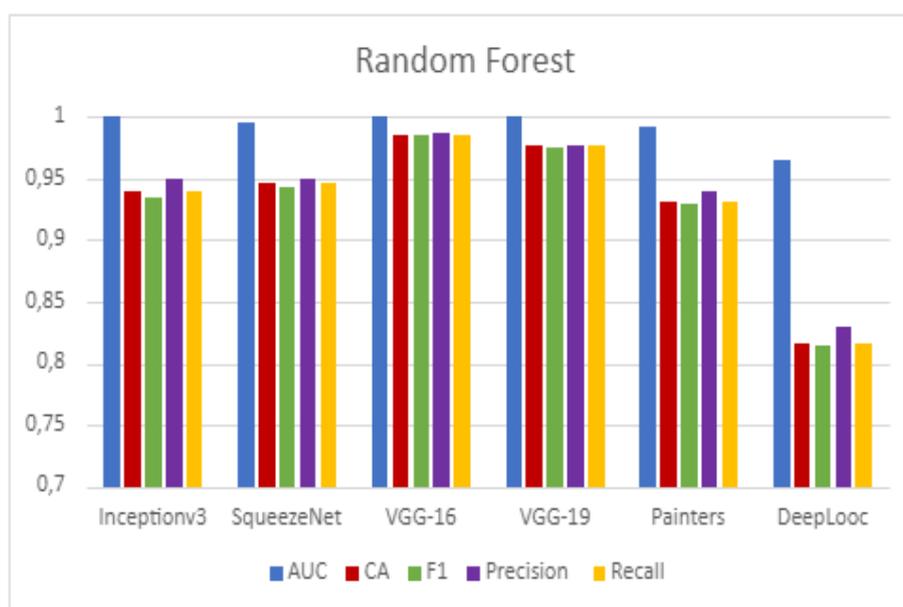
Random Forest									
Incorporador	Nº tress	Nº atribuSplit	limitDepth	do not split	AUC	CA	F1	Precision	Recall
VGG-16	150	20	20	5	1	0.993	0.993	0.993	0.993
VGG-16	150	20	30	5	1	0.993	0.993	0.993	0.993
VGG-16	150	20	40	5	1	0.993	0.993	0.993	0.993
VGG-16	150	20	50	5	1	0.993	0.993	0.993	0.993
VGG-16	150	40	20	5	1	0.993	0.993	0.993	0.993

Fonte: Produção do autor (2021).

No modelo Random Forest se variação os parâmetros, o número de árvore de decisão a ser incluído na árvore que varia entre 50, 100 e 150, o número de árvore a cada split de 20, 30, 40 e 50, o limite da profundidade da arvore individual entre 20, 30, 40 e 50 e o número que limita a divisão se manteve em 5.

Os cinco melhores resultados do modelo *Random Forest* são dispostos na Tabela 8, o teste com VGG-16 converge em 1 para AUC e para as demais métricas se mantem em 99%.

Figura 20 - Gráfico do modelo Random Forest Abordagem 2



Fonte: Autoria própria (2021).

A rede convolucional VGG-16 demonstra os melhores resultados para o modelo *Random Forest* como é ilustrado na figura 20, seguido pela a *VGG-19*, *SqueezeNet*, *Inceptionv3*, *Painters* e com o desempenho mais baixo o *DeepLooc*.

Figura 21 - Matriz de confusão do modelo Logic Regression

		Predicted				Σ
		arvore	carro	casa	predio	
Actual	arvore	100.0 %	0.0 %	0.0 %	0.0 %	41
	carro	0.0 %	100.0 %	0.0 %	0.0 %	40
	casa	0.0 %	0.0 %	97.6 %	0.0 %	40
	predio	0.0 %	0.0 %	2.4 %	100.0 %	26
Σ		41	40	41	25	147

Fonte: Universidade de Ljubljana (2021).

Após a análise dos resultados da técnica *Logic Regression* é possível notar que a técnica não teve dificuldade em classificar as classes árvore, carro e prédio já na classe casa teve um acerto de 97.6% o modelo confundiu com a classe prédio com erro de 2.4%, como é ilustrado na Figura 21 A matriz foi gerada através do melhor valor obtido no treinamento.

Figura 22 - Imagens classificadas erradas pelo modelo Random Forest



predio

Fonte: Fonte: Autoria própria (2021).

O modelo *Random Forest* errou somente uma imagem como mostrado na Figura 23 sendo mostrada uma casa e foi classificada como prédio.

Tabela 9 - Resultados do Modelo AdaBoost abordagem 2

AdaBoost									
Incorporador	ClassAlgorit	funct/regres	Nº Estimato	Lear_rate	AUC	CA	F1	Precisio	Recall
VGG-16	SAMME	Linear	20	1	0.950	0.925	0.925	0.926	0.925
VGG-16	SAMME	Square	20	1	0.950	0.925	0.925	0.926	0.925
VGG-16	SAMME	Exponencial	20	1	0.950	0.925	0.925	0.926	0.925
VGG-16	SAMME.R	Linear	20	1	0.950	0.925	0.925	0.926	0.925
VGG-16	SAMME.R	Square	20	1	0.950	0.925	0.925	0.926	0.925

Fonte: Produção do autor (2021).

O treinamento com *AdaBoost* teve os parâmetros variados, em algoritmo de classificação *SAMME* e *SAMME.R*, função de perda de regressão *Linear*, *square* e *exponential* e número de estimadores 20, 30, 40, 50 e 100 e a taxa de aprendizagem se manteve em um. É possível notar na Tabela 9 que os valores convergem em AUC à área sobre a curva em 95%, o CA que é a precisão da classificação, o F1 que é a média harmônica entre *Precision* e *Recall*, a *Precision* que é a precisão de classificação correta e o *Recall* sendo a proporção de verdadeiro positivo em 92%.

Figura 23 - Matriz de confusão modelo AdaBoost

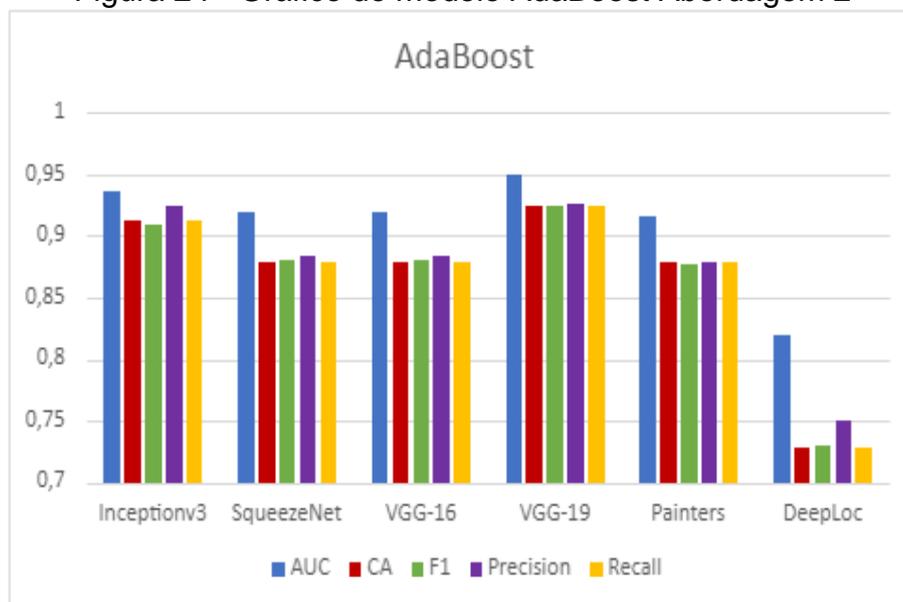
		Predicted				Σ
		arvore	carro	casa	predio	
Actual	arvore	93.0 %	2.4 %	0.0 %	0.0 %	41
	carro	0.0 %	97.6 %	0.0 %	0.0 %	40
	casa	2.3 %	0.0 %	94.4 %	18.5 %	40
	predio	4.7 %	0.0 %	5.6 %	81.5 %	26
Σ		43	41	36	27	147

Fonte: Universidade de Ljubljana (2021).

o modelo *Adaboost* é a técnica com maior quantidade de erro, isso é possível ser visto na Figura 24 onde o modelo confundiu as classes de carro com arvore 2.4%, arvore com casa

2.3%, predio com arvore 4.7%, casa com predio 5.6% e a maior quantidade de erro com a classe predio que foi confundida com casa 18.5% de erro na classificação das imagens.

Figura 24 - Gráfico do modelo AdaBoost Abordagem 2



Fonte: Autoria própria (2021).

Através dos resultados obtidos no treinamento da técnica *AdaBoost* é possível visualizar que a técnicas de aprendizagem de máquina VGG-19 retém os melhores resultado em ambos valores AUC, CA, F1, *Precision* e *Recall* como é ilustrado Figura 24 na, seguido pelo Inceptionv3 com o segundo melhor resultado, o *Painters*, e tanto o *SqueezeNet* e VGG-16 tiveram os valores bem próximos, e com o resultado mais baixo o *DeepLoc*.

6 CONCLUSÃO

Neste relatório, foram apresentadas as atividades desenvolvidas no período de 01 de setembro de 2020 a 31 de julho de 2021, referente ao projeto “Emprego de Inteligência Artificial na escolha automática de algoritmos e parâmetros de técnicas de processamento de imagens obtidas por Drones para aplicação no sensoriamento remoto”. Após a análise dos resultados, foi possível notar que alguns dos modelos demonstram bons resultados dependendo do tipo de dado e que, na maioria dos casos, a Rede Neural Artificial teve um resultado superior em relação aos demais modelos. Portanto, pode-se dizer as atividades do projeto foram desenvolvidas de forma bastante satisfatória nesse período.

REFERÊNCIAS

UNIVERSITY OF LJUBLJANA (Eslovénia) *et al*, (org.). **Orange**: Software de machine learning e visualização de dados. Versão 3.27.0. [S. l.], 2020. Disponível em: <https://orange.biolab.si/download/#windows>. Acesso em: 26 ago. 2020.

PYTHON (Wilmington, Delaware, EUA) (ed.). Linguagem Programação Python. *In: Python*. 3.9. [S. l.], 2021. Disponível em: <https://www.python.org/>. Acesso em: 19 out. 2021.

CINERGIX PTY LTD (Austrália) *et al*, (org.). **Creately**: software de diagrama e fluxograma. Versão 8.2.14. [S. l.], 2008. Disponível em: <https://creately.com/>. Acesso em: 27 ago. 2020.

Paulino, Ângelo; Guimarães, Lamartine ;Shiguemori, Elcio. **Assessment of Noise Impact on Hybrid Adaptive Computational Intelligence Multisensor Data Fusion Applied to Real-Time UAV Autonomous Navigation**. 2019. IEEE Latin America Transactions. 18. 295-302. 10.1109/TLA.2020.9085283.

Roos, Daniel; Lorena, Ana; Shiguemori, Elcio. **Comparing ORB and AKAZE for visual odometry of unmanned aerial vehicles**. In: Conference of Computational Interdisciplinary Science, 2016, São José dos Campos. Proceedings of Conference of Computational Interdisciplinary Science. São José dos Campos: INPE, 2016. v. 4. p. 121.

LACERDA, Marielcio Gonçalves. **Abordagem GEOBIA para imagens VHR obtidas por aeronaves remotamente pilotadas e sensores satelitais com o uso de classificadores individuais e ensemble**. 2020. 192 p. Tese (Mestrado em Ciências) - Instituto Tecnológico de Aeronáutica, São José Dos Campos, 2020.

FORNARI, Gabriel. **Caracterização e desenvolvimento de um sistema autoadaptativo para estimação da posição de veículos aéreos não tripulados baseado em imagens**. 2020. 183 p. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisa Espaciais, São José Dos Campos, 2020.

WANG, Jason; PEREZ, Luis. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. **Computer Vision and Pattern Recognition**, <https://arxiv.org/abs/1712.04621>, 13 dez. 2017.

Y Freund and R. E. Schapire. **Game theory, on-line prediction and boosting**. In **Proceedings of the Ninth Annual Conference on Computational Learning Theory**, pages 325–332, 1996.

HURWITZ, Judith; KIRSCH, Daniel. **Machine Learning for dummies a willey Band**. Hoboken: Wiley, 2018. 75 p.

Spyder IDE. <https://www.spyder-ide.org/>. Acessado em 28/08/2020.

Scikit-Learn. Disponível em: <https://scikit-learn.org/stable/>. Acessado em 10/08/2021.

OpenCv. Disponível em: <https://opencv.org/> . Acessado em 18/08/2021.

Google Cloud. <https://cloud.google.com/tpu/docs/tpus>. Acessado em 20/04/2021.

Keras. Disponível em: <https://keras.io/>. Acesso em: 18 ago. 2021.

TENSORFLOW. *In: TensorFlow*. 2.6. [S. l.], 2015. Disponível em: <https://www.tensorflow.org/?hl=pt-br>. Acesso em: 18 ago. 2021.

Orange Data Mining. <https://orangedatamining.com/widget-catalog/data/file/> Acessado em 18/08/2021.

GeoInfo. Disponível em: www.geoinfo.info/geoinfo2021/. Acessado em 10/08/2021.

ImageNet. Disponível em: <https://www.image-net.org/>. Acessado em 18/08/2021.

Rosenblatt, F. **The perceptron: A probabilistic model for information storage and organization in the brain.**(Psychological Review) (1958).

Hosmer, David W.; Stanley Lemeshow.**Applied Logistic Regression**, 2nd ed. [S.l.]: New York. Chichester, Wiley(2000).

ZAMBIASI, Saulo Popov. **O Neurônio Artificial**. [S. l.], 2008. Disponível em: https://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/index.html. Acesso em: 19 ago. 2021.

Ho, Tin Kam. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal (1995). 282p.

IANDOLA, Forrest n. *et al.* SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. **Computer Vision and Pattern Recognition (cs.CV); Artificial Intelligence (cs.AI)**, [s. l.], 4 nov. 2016.

SZEGEDY, Christian *et al.* Rethinking the Inception Architecture for Computer Vision. **Computer Vision and Pattern Recognition (cs.CV)**, [s. l.], 2 dez. 2015.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. **Computer Vision and Pattern Recognition (cs.CV)**, [s. l.], 10 abr. 2015.

SØNDERBY, Casper Kaae *et al.* DeepLoc: prediction of protein subcellular localization using deep learning. **Bioinformática**, [s. l.], 1 nov. 2017.

Ilenič, :Nejc; **Painter by Numbers Competition**. 11 Nov 2016.