UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Sorocaba

PROGRAMA DE PÓS-GRADUAÇÃO em
Ciências Ambientais

CLAUDIA LILIANA GUTIERREZ ROSAS

ANÁLISE DA MORFOMETRIA DOS AGREGADOS DO SOLO UTILIZANDO IMAGENS DIGITAIS POR MEIO DE REDES NEURAIS

Sorocaba

2021

CLAUDIA LILIANA GUTIERREZ ROSAS

ANÁLISE DA MORFOMETRIA DOS AGREGADOS DO SOLO UTILIZANDO IMAGENS DIGITAIS POR MEIO DE REDES NEURAIS

Dissertação apresentada como requisito para a obtenção do título de Mestre em Ciências Ambientais da Universidade Estadual Paulista "Júlio de Mesquita Filho" na Área de Concentração Diagnóstico, Tratamento e Recuperação Ambiental

Orientador: Admilson Irio Ribeiro

Coorientador: Ivando Severino Diniz

Sorocaba

2021

R789a

Rosas, Claudia Liliana Gutierrez

Análise da Morfometria dos Agregados do Solo Utilizando Imagens Digitais por Meio de Redes Neurais / Claudia Liliana Gutierrez Rosas. -- Sorocaba, 2021

71 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Admilson Irio Ribeiro

Coorientador: Ivando Severino Diniz

1. Redes neurais artificiais. 2. Agregado do solo. 3. Morfometria dos agregados do solo. 4. Imagens digitais. 5. Rede neural convolucional. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba.
Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Sorocaba

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: ANÁLISE DA MORFOMETRIA DOS AGREGADOS DO SOLO UTILIZANDO IMAGENS DIGITAIS POR MEIO DE REDES NEURAIS

AUTORA: CLAUDIA LILIANA GUTIERREZ ROSAS

ORIENTADOR: ADMILSON IRIO RIBEIRO

COORIENTADOR: IVANDO SEVERINO DINIZ

Aprovada como parte das exigências para obtenção do Título de Mestra em CIÊNCIAS AMBIENTAIS, área: Diagnóstico, Tratamento e Recuperação Ambiental pela Comissão Examinadora:

Prof. Dr. ADMILSON IRIO RIBEIRO (Participaçao Virtual)
Engenharia Ambiental / Unesp - ICT Sorocaba

Prof. Dr. ROGER MANUEL MESTAS VALERO (Participaçao Virtual)
Ingeniería Agraria / Universidad Catolica Sedes Sapientiae

Prof. Dr. FELIPE HASHIMOTO FENGLER (Participaçao Virtual)
Engenharia Civil / Faculdade de Engenharia de Sorocaba (FACENS)

Sorocaba, 26 de março de 2021

A minha mãe Sofía Rosas Colonia, pela força, pelo amor, pelo carinho, mesmo estando longe, sentia sua presença perto de mim, não poderei pagar todo o esforço que fez, faz e continua fazendo por mim.

AGRADECIMENTOS

Gratidão é sentir-se afortunada e em abundância, cercado por pessoas maravilhosas que têm sido fundamentais para alcançar esse objetivo, e dedico essa conquista a todos nós, familiares, amigos e professores. Agradecer especialmente:

À *Universidade Estadual Paulista* e ao *Programa de Pós-Graduação em Ciências Ambientais* do Instituto de Ciência e Tecnologia, campus de Sorocaba, Brasil.

Ao *Programa Nacional de Becas y Crédito Educativo (PRONABEC, Beca Presidente de la República - Perú)*, pelo amparo financeiro à pesquisa durante o desenvolvimento da tese.

Aos amigos e familiares, em especial aos meus pais *Sofia Rosas Colonia* e *Alberto Gutierrez Colonia*, pessoas de superação e coragem, fonte da minha motivação.

Aos professores, *Dr. Admilson Irio Ribeiro* e *Dr. Ivando Severino Diniz*, pela orientação por todos os ensinamentos, mas principalmente pelo apoio e amizade.

Aos colegas da Faculdade, em especial para a colega *María Fernando Tejada Begazo*, quem esteve comigo nas dificuldades e desafios do desenvolvimento da pesquisa.

E por último, mas não menos importante a meu namorado *Bruno Lima Bezerra*, pelo apoio, e por estar ao meu lado dando força e motivação e enfrentar todos os desafios pessoais, acadêmicos e profissionais.

RESUMO

A estrutura do solo consiste em uma das propriedades físicas mais sensíveis ao manejo e pode ser afetada pelo uso intenso de máquinas e aplicação regular de insumos. A perda estrutural do solo pode ser avaliada por diferentes metodologias, no entanto a análise de agregados e suas diferentes formas mostram-se de fácil interpretação e expedita, mas, a avaliação estrutural por meio dos agregados ainda requer desenvolvimento de pesquisas. Dentro dos conceitos de agricultura de precisão buscou-se nessa proposta o desenvolvimento e avaliação de um modelo de rede neural artificial capaz de detectar e classificar diferentes formas de agregados do solo. A metodologia desenvolvida utilizou-se da linguagem de programação "*Python3*" com a biblioteca "Keras" e "*TensorFlow*", utilizando uma Rede Neural Convolucional, com aplicação de transferência de aprendizagem da arquitetura *MobileNetV2* em três experimentos, Experimento 1 com 5 classes morfométricas (prismática, angular, subangular, arredondada e redonda), Experimento 2 com 4 classes (prismática, angular, subangular e redondado) e Experimento 3 com 3 classes (prismática, angular, redondado). Durante o desenvolvimento da rede e tratamento da *data_set*, foram utilizados os métodos de *data_augmentation, Under sampling* e *data cleaning.* Treinando assim uma da rede neural capaz de identificar as diferenças e similaridades das classes morfométricas dos agregados por meio de imagens, visando aplicações em sistemas de precisão e inteligentes para a manutenção estrutural dos solos. Os resultados do treinamento da *MobileNetV2* dos três experimentos mostraram uma *accuracy* média de 79%, sendo o Experimento 3 com melhor desempenho com uma *accuracy* de 87%. Para os três experimentos, as classes morfométricas de maior precisão foram as arredondadas, redondo e o angular, enquanto, para as classes prismática e subangular a rede apresentou capacidade de detecção e classificação errática. Conclui-se, que a preparação da *data_set* é uma fase muito importante, pode influenciar e fazer diferença no desempenho da rede, possuir um conjunto de dados robusto aos ajustes, a qualidade e quantidade de imagens por classe, o pré-processamento em geral, além do hardware, software e o nível formação na matéria da pesquisa. A concepção da aplicação de aprendizado de máquinas para a avaliação estrutural dos solos mostra-se promissora. Essa condição denota que esse trabalho não se constitui em um produto final “acabado” e pelo contrário constitui-se em um ponto de partida para pesquisas, desenvolvimento e inovação na gestão ambiental da qualidade estrutural dos solos em processos produtivos agrícolas.

Palavras-chave: Agregado, do solo, Redes neurais artificiais, Morfometria, imagens

ABSTRACT

Soil structure is one of the most sensitive physical properties to management and can be affected by intense use of machinery and regular application of inputs. The structural loss of the soil can be evaluated by different methodologies; however, the aggregates analysis and its different forms are easy to interpret and expedite, but the structural evaluation through the aggregates still requires research development. Within the concepts of precision agriculture, this proposal sought to develop and evaluate an artificial neural network model capable of detecting and classifying different forms of soil aggregates. The methodology developed used the programming language "Python3" with the library "Keras" and "TensorFlow", using a Convolutional Neural Network, with learning transfer application of the MobileNetV2 architecture in three experiments, Experiment 1 with 5 morphometric classes (prismatic, angular, subangular, rounded and rounded), Experiment 2 with 4 classes (prismatic, angular, subangular and rounded) and Experiment 3 with 3 classes (prismatic, angular, rounded). During the development of the network and treatment of data_set, the methods of data_augmentation, under sampling and data cleaning were used. Thus, training a neural network capable of identifying the differences and similarities of the morphometric classes of aggregates through images, aiming at applications in precision and intelligent systems for the structural maintenance of soils. The MobileNetV2 training results from the three experiments showed an average accuracy of 79%, with Experiment 3 having the best performance with an accuracy of 87%. For the three experiments, the morphometric classes with the highest accuracy were rounded, round and angular, while for the prismatic and sub-angular classes the network showed erratic detection and classification capabilities. It is concluded, that the preparation of the data_set is a very important phase, can influence and make a difference in the performance of the network, having a dataset robust to the adjustments, the quality and quantity of images per class, the preprocessing in general, besides the hardware, software and the level of training in the research subject. The design of the application of machine learning for the structural assessment of soils shows promise. This condition denotes that this work is not a "finished" final product, but rather a starting point for research, development, and innovation in the environmental management of the structural quality of soils in agricultural production processes.

Keywords: Soil Aggregate, Artificial Neural Networks, Morphometry, Images

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

AR =Área

Ard =Arredondamento

ASP =Aspecto

AUROC =Area under the Receiver Operating Characteristic

BPN =back propagation neural network BDT - binary decision tree

CNN =Rede neural convolucional

CMP =Compacidade

DF =Diâmetro Feret

DSM =Mapeamento Digital De Solo

FAO =United Nations Food Organization

MLR =Regressão Linear Múltipla

MWD =Diâmetro Ponderado Médio

MLP =Perceptron multicamadas

PER = Perímetro

RNA = Redes Neurais Artificiais

ReLU =Rectified Linear Unit

ROC =Receiver Operating Characteristic

RUG =Rugosidade

1. SUMÁRIO

1. INTRODUÇÃO

A produção agrícola convencional consiste numa atividade antrópica impulsionada principalmente pelo crescimento populacional e econômico; sendo uma atividade necessária para produção de alimentos e a manutenção das populações. No mundo coexistem diferentes formas de fazer agricultura, no entanto, a agricultura contemporânea se apresenta em grande parte de forma intensiva, promovendo mudanças significativas nas características físicas, químicas e biológicas do solo (Anghinoni, Carvalho & Costa; 2013).

Nesse sentido, as alterações estruturais e funcionais do solo condicionam sua produtividade, sendo necessário seu estudo e quantificação, por meio de indicadores que representam as diferentes condições da qualidade dos solos (RABOT *et al*., 2018; ALEMANHA, 2017). Dentro do universo das variáveis indicadoras da qualidade do solo, encontra-se a condição estrutural (disposição dos agregados). A análise correta dessa variável indicadora, pode levar a uma melhor gestão da qualidade física do solo, que por sua vez influencia o crescimento das plantas e na recuperação do solo degradado (AVLIS, 2014).

A condição dos agregados do solo interfere em sua estabilidade, facilidade de penetração da raiz, aeração, capacidade de drenagem, armazenamento de água, plasticidade e retenção de nutrientes (RUCKS *et al*., 2004). A agregação não pode ser considerada uma mera reação física, química ou biológica, mas sim o reflexo da interação de todas as forças atuantes no solo. Ela pode ser considerada um indicador confiável da saúde, condição geral e qualidade do solo (PECHE FILHO, 2018).

Nesse escopo, a condição estrutural do solo pode ser determinada pelo grau de agregação das partículas primarias do solo, uma das variáveis para mensurar esse grau de agregação consiste na estabilidade de seus agregados, bem como a classe dos tamanhos e formas (FAO, 2016). A análise dos agregados e suas diferentes formas se mostram de fácil interpretação e expedita, porém, a avaliação estrutural por meio dos agregados ainda requer pesquisas e desenvolvimento solo (PECHE FILHO, 2018).

Nos últimos anos têm ocorrido dentro das ciências ambientais investigações sobre o uso de sistemas compactos e portáteis para auxiliar o monitoramento e a avaliação ambiental. Esses sistemas são desenvolvidos em universidades ou centros de pesquisa em todo o mundo estão, cada vez mais utilizando mais as tecnologias de informação e comunicação, incluindo tecnologias baseadas em inteligência artificial (RUBIO et al. 2016; VILA & PENIN, 2007, MATICH, 2001).

O desenvolvimento de sistemas inteligentes, como o aprendizado de máquinas que se baseiam nos métodos de análise visual, são capazes de auxiliar na gestão do solo (VAN LEEUWEN *et al.*, 2018; BALL *et al*., 2017). Nessa inserção estão as Redes Neurais Convolucionais (CNN) que consistem em um algoritmo de Aprendizado Profundo (*Deep learning*) podendo ser utilizadas na classificação de imagens, reconhecendo as características únicas de cada objeto e generalizando a informação a partir de conjuntos de dados (QUINTERO *et al*., 2018). As CNN são um tipo de Redes Neurais Artificiais (RNA) que se baseiam em um modelo matemático inspirado no funcionamento das redes neurais biológicas; desenvolvidas com o objetivo de criar sistemas que aprendem automaticamente com capacidade de reconhecimento de padrões complexos (TABLADA & TORRES, 2015; IZAURIETA & SAAVEDRA, 2000; SARMIENTO-RAMOS, 2020).

Partindo de um banco de dados que foi utilizado para uma avaliação visual dos agregados do solo e dentro dos conceitos e aplicabilidade de redes neurais convolucionais buscou-se o desenvolvimento de um modelo de rede neural artificial para detectar e classificar as diferentes formas de agregados de solo visando aplicação em sistemas de conservação de precisão.

2. OBJETIVOS

2.1 Objetivo geral

- Classificar os agregados do solo através de variáveis morfométricas utilizando uma rede neural artificial

2.2 Objetivos específicos

- Levantamento das propriedades morfometrias e preparação da data *set* (imagens para treinamento, validação e teste).
- Transferência de aprendizagem para o desenvolvimento da rede neural para classificação da forma dos agregados.
- Avaliação do desempenho do aprendizado da rede neural desenvolvida

3. REVISÃO CONCEITUAL

3.1 Agregação do solo

3.1.1 Solo

O solo pode ser definido como a camada mais superficial da crosta terrestre, um dos recursos naturais mais importantes porque é o substrato que sustenta a vida no planeta.

O solo considerado com uma mistura de substâncias líquidas (água), gasosas (ar), e estruturas sólidas (agregados, raízes, organismos macro/microscópicos vivos e material orgânico em diferentes estágios de decomposição) (MOREIRA & SIQUEIRA, 2006).

O solo também pode ser considerado um elemento natural dinâmico e vivo que é a interface entre a atmosfera, a litosfera, a biosfera e a hidrosfera, onde há uma contínua troca de matéria e energia. Portanto, o solo tem a capacidade de desenvolver uma série de funções essenciais na natureza ambiental, ecológica, econômica, social e cultural (ORTIZ *et al*., 2007).

O solo não consiste apenas em uma coleção de fragmentos de rocha, ele atua como um meio de apoio ao crescimento da planta e abriga uma grande variedade de organismos de grandes a muito pequenos de escala microscópica (BRADY & WEIL, 2009).

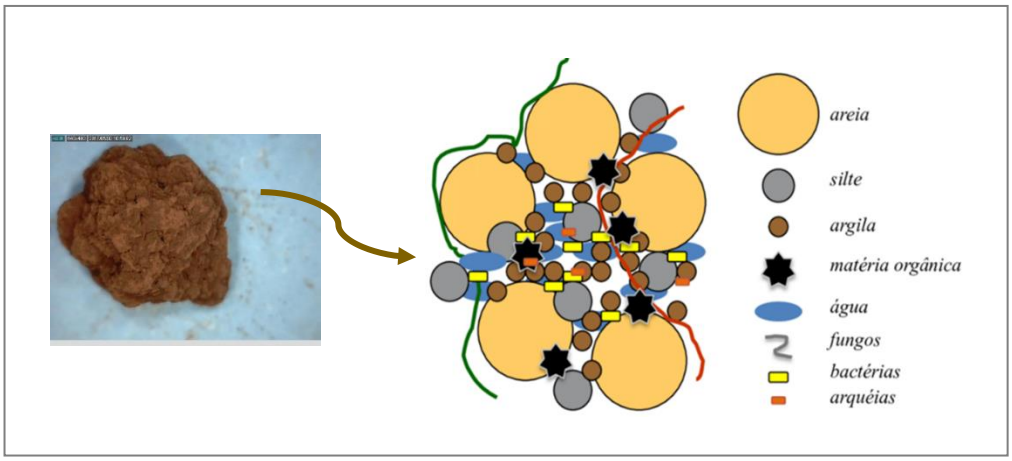
3.1.2 Estrutura e agregação do solo

A estrutura se refere ao padrão de arranjo das partículas primárias (areia, silte e argila) em unidades estruturais (peds ou agregados). As forças de adesão e coesão e os agentes

cimentantes são responsáveis pela união das partículas primárias (Figura1) (BRADY & WEIL, 2009; RUCKS *et al*., 2004). Os agregados são principalmente aqueles que estruturam o perfil do solo, influenciando nas funções básicas, tais como infiltração, aeração, armazenamento de água, retenção de nutrientes e desenvolvimento radicular (PECHE FILHO, 2018).

A estrutura do solo assume uma forma em base a suas propriedades físicas e químicas, devido a isso, a estrutura é um dos principais componentes da fertilidade do solo, solos com boa estrutura, possuem valores adequados de porosidade, com boa aeração, infiltração e retenção de água (RALISCH *et al*., 2017).

Figura 1- Ilustração de um agregado



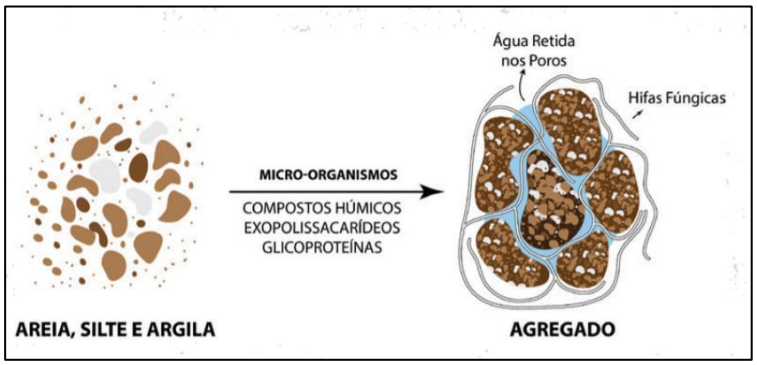
Fonte: Adaptado de (BINI et al., 2016)

Os agregados de solo são formados através da atividade física, química e biológica do solo (INFOAGRO, 2019). Também, encontrassem associada com a formação dos micro-hábitats, devido que funcionam como um suporte físico para a aderência microbiana e proporcionam condições diferenciadas de aeração e disponibilidade de nutrientes (BINI *et al*., 2016).

A formação de agregados pode ser definida como um processo complexo e pode ser influenciada por fatores humanos como a lavoura, o caminhar na superfície, as estradas e até mesmo o pastoreio. Para a formação de agregados no solo, são necessárias duas condições, primeiro uma força mecânica de algum tipo, que deve aproximar as partículas, o segundo, um agente cimentante, que deve estar disponível no meio para consolidar esta ligação e gerar o agregado, alguns exemplos de agentes cimentícios incluem matéria orgânica e materiais de cal, tais como carbonato de cálcio (PASSARIN *et al*., 2007).

No caso da matéria orgânica, microrganismos do solo e fauna presentes (minhocas, etc.), no processo de decomposição, secretam compostos orgânicos que servem como "cola" para produzir a colagem e cimentação dos agregados (BINI et al., 2016). Outro agente que ajuda na formação de agregados, são as raízes das plantas, através da secreção de compostos orgânicos chamados exsudados radiculares, que ajudam a unir o solo próximo à zona radicular. Como é mostrado na Figura 2, as hifas fúngicas também contribuem para a formação e estabilidade dos agregados, enredando e tecendo em torno de partículas do solo (DE CARVALHO *et al*., 2017; CASTRO FILHO *et al*., 1998; CESAR SALTON *et al*., 2008).

Figura 2 - Formação dos agregados através da ação de micro-organismos



Fonte: (de Carvalho et al., 2017)

O tipo e a quantidade de minerais argilosos presentes desempenham um papel influente na formação da agregação (INFOAGRO, 2019). Os solos bem agregados oferecem melhores condições para o desenvolvimento das plantas, daí a importância de obter informações sobre os agregados do solo, como tamanho, forma e estabilidade (PASSARIN et al., 2007).

3.1.3 Estabilidade dos agregados

A estabilidade dos agregados do solo pode ser o resultado da ação mecânica da ligação das células e hifas dos organismos, dos efeitos cimentícios dos produtos de síntese microbiana e dos produtos de decomposição (matéria orgânica) (WOHLENBERG et al., 2004). A matéria orgânica e outras substâncias atuam como cimento entre as partículas individuais que formam o solo e os agregados (BRADY & WEIL 2009).

Algumas glicoproteínas produzidas pelos fungos micorrizos, como a glomalina (Kochem et al., 2016), podem contribuir na estabilidade dos agregados, existe correlação positivas entre a glomalina e o carbono, o nitrogênio e a agregação do solo (DA SILVA et al., 2016).

A agregação do solo, pode ser avaliada por meio da estabilidade dos agregados em água e resistência ao impacto da gota (CASTRO FILHO et al., 1998). A estabilidade do agregado desempenha um papel fundamental na erosão do solo (GUERRA, DA SILVA & BOTELHO, 2009).

Da mesma forma, GUERRA et al. (2009) explicam se a estabilidade dos agregados for baixa, os agregados se quebram facilmente e podem crustar a superfície do solo, dificultando a infiltração e aumentando o escoamento superficial. Enquanto agregados com estabilidade que estão na água, dificultam a quebra e separação das partículas no solo, impedindo a impermeabilização da superfície e mantendo a infiltração alta no topo do solo. A consistência e estabilidade dos agregados será extremamente importante para definir a taxa de erosão que pode afetar um determinado solo.

A agregação do solo pode sofrer mudanças permanentes ou temporárias causadas pelas práticas de manejo do solo e da cultura (WOHLENBERG et al., 2004). Entre as variáveis que afetam a estrutura do solo, a matéria orgânica pode ser o fator mais importante na formação e estabilidade dos agregados, e melhorar a estabilidade da estrutura do solo resulta em maior resistência à erosão e maior capacidade de retenção de umidade (MARIA et al., 2007).

PECHE FILHO (2018), indica que a variabilidade da agregação de referência consiste em compreender o comportamento dos agregados em condições naturais, ou seja, em condições de referência. Levando em conta as características de forma, superfície e biogênese que precedem o início das intervenções de gestão. A análise da variabilidade da agregação natural nos permite entender como o solo responde ao processo de agregação em condições naturais formando a estrutura do solo (Figura 3).

Figura 3 - Variabilidade da agregação com a estrutura do solo

Fonte: (RALISCH et al., 2017)

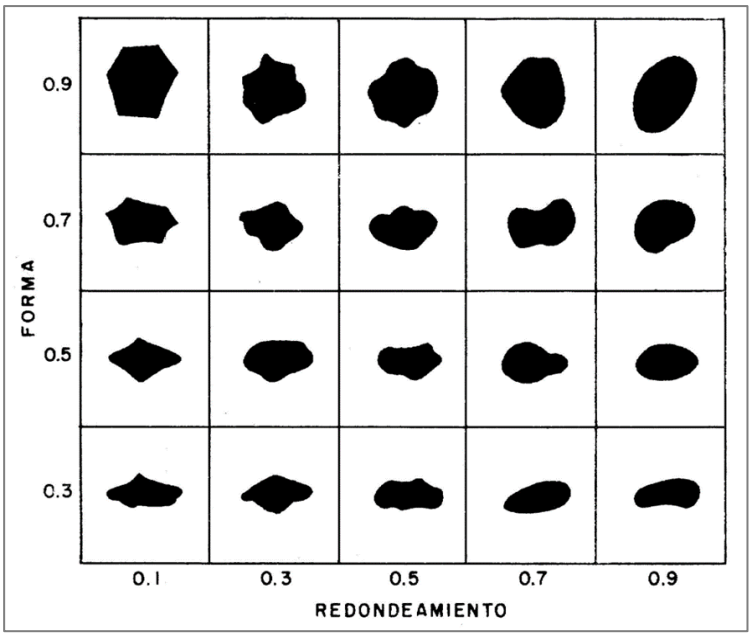
Por outro lado, WOHLENBERG et al. (2004), indicam que as espécies de cobertura vegetal, juntamente com as práticas de manejo e conservação, recuperam ou mantêm as características físicas do solo, tais como agregação. Há uma ação direta das culturas sobre a formação e estabilização dos agregados, é melhor a estabilidade e distribuição de tamanho de agregados maiores em sistemas de cultivo que fornecem matéria orgânica e cobrem o solo durante todo o ano (WOHLENBERG et al., 2004; BRADY & WEIL, 2009).

3.1.4 Morfometría dos agregados

A estrutura do solo consiste no arranjo das partículas primárias (areia, silte e argila, que juntos formam os agregados. Outro conceito definido por GUERRA et al. (2009), descreve a estrutura dos solos como uma unidade estrutural com um conjunto coerente de partículas primárias do solo com forma e tamanho definidos. Assim, podem existir diferentes tipos de agregados dentro do mesmo horizonte de solo, a presença destes pode variar tanto horizontal como verticalmente.

A organização hierárquica do solo, possui cinco grupos de parâmetros: morfométricos, geométricos, físicos, químicos e energéticos (GERVASIO PEREIRA et al., 2019). A nível morfometríco, os agregados do solo vão do nível micro (menos de 0,25 mm de diâmetro) ao nível macro (mais de 0,25 mm de diâmetro). Além disso, como se observa na Figura 4, os agregados eles podem se assemelhar a várias formas: Arredondado, redondo, prismático, angular (SALTON et al., 2008; STEINBECK, 2006). Estas formas variadas permitem que o solo saudável tenha espaços porosos para o ar e a água, que são necessários para o crescimento saudável das plantas (CASTRO FILHO et al., 1998; CESAR SALTON et al., 2008).

Figura 4 - Formas variadas de agregados



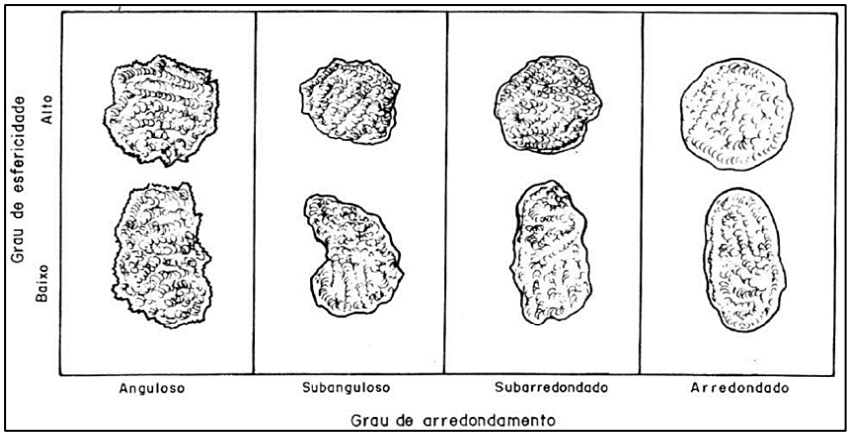
Fonte: (Durafour et al., 2015)

FAUSTINO (2018) e KOHN (2017), em suas pesquisas, citam vários autores onde indicam que a análise morfométrica dos agregados é importante, pois a geometria dos agregados interfere na distribuição do diâmetro dos poros, o que modifica a dinâmica dos nutrientes do ar, da água e do solo e, consequentemente, afeta o crescimento das raízes das plantas.

De acordo com os agregados de solo são descritos por sua forma, como:

A) Arredondado: Eles podem ter a estrutura da forma: Granular, pequenos agregados, geralmente com menos de 2 cm, bem arredondados. A presença de estruturas arredondadas significa um ambiente poroso onde água, ar e vida animal e vegetal circulam livremente. São estáveis na água, indicando boa resistência à erosão (GUERRA, et al. 2009). Podem existir diferentes graus de esfericidade como é observado na Figura 5, assim determinar um agregado arredondado.

Figura 5 - Graus de esfericidade



Fonte: (Catafesta, 2014)

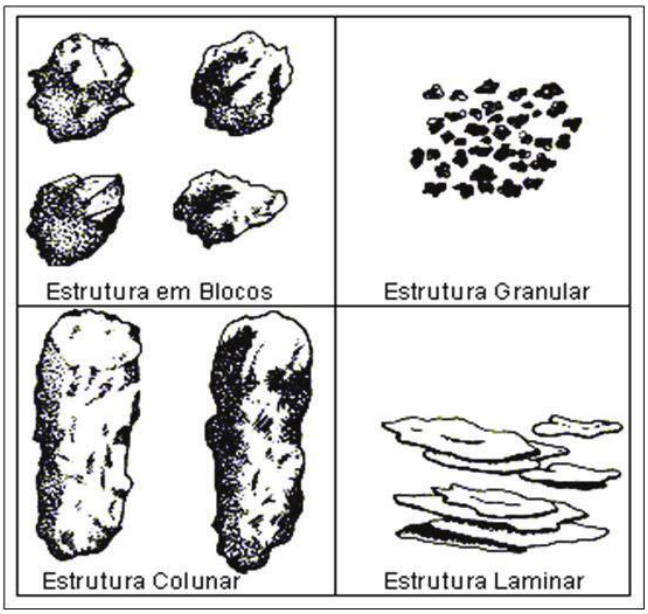
B) Blocos poliédricos ou subangulares: agregados maiores com faces planas, arredondadas, com cantos arredondados, quase quadrado ou angular com bordas mais ou menos pronunciadas. Eles são normalmente encontrados no horizonte B, onde há acúmulo de argila. Os blocos relativamente grandes indicam que o solo resiste à penetração e ao movimento da água (SALTON et al., 2008).

C) Angular: os agregados têm faces planas e vértices formados por ângulos. Os agregados angulares são mais compactos e restringem a atividade biológica, a água e o ar circulam através das fendas entre os agregados, mas são restritos (FAUSTINO, 2018; FAO, 2009).

D) Prismático: o desenvolvimento dos agregados é mais visível verticalmente. As estruturas prismáticas correspondem àqueles agregados cuja dimensão vertical é maior do que a dimensão horizontal. Os solos com agregados prismáticos, a água circula com maior dificuldade e a drenagem é deficiente (FAO, 2009).

A forma dos agregados pode influenciar no tipo e na formação da estrutura do solo, na figura 6 se observa os diferentes tipos de estrutura do solo.

Figura 6 - Formas das unidades estruturais do solo



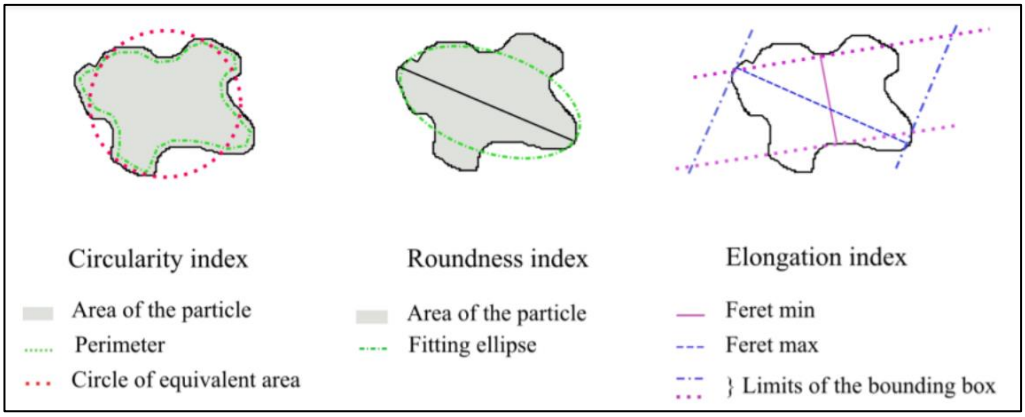
Fonte: (Silva *et. al*, 2015)

3.1.5 Análise Morfométrica dos Agregados através da análise de imagem

De acordo com FAUSTINO (2018), a análise morfométrica dos agregados é importante, pois a geometria dos agregados interfere na distribuição do diâmetro dos poros, o que modifica a dinâmica dos nutrientes do ar, da água e do solo e, consequentemente, afeta o crescimento das raízes das plantas.

Existem diferentes maneiras de avaliar a morfometria dos agregados principalmente desenvolvida na pedologia e engenharia civil (León & Ramírez, 2010), A análise morfométrica dos agregados consiste basicamente em três etapas (FAUIS-TINO, 2018, SOUZA et al. 2018, PECHE FILHO, 2018), na aquisição das imagens por scanner ou câmeras; depois na análise das imagens para obtenção de características ou parâmetros morfométricos (Figura 7); e finalmente na interpretação dos resultados gerados.

Figura 7 - Obtenção de características de forma de um agregado



Fonte: (Durafour et al., 2015)

Para FAUSTINO (2018), o estudo da morfometria do agregado consiste em avaliar a seguintes características: Área, que é medida com o número de pixels no polígono e indica o estado de agregação do solo; Perímetro, que é o comprimento da projeção do limite externo do agregado, que está diretamente relacionado com a área dos agregados; Aspecto, que expressa a forma do agregado, que dá um resultado entre 0 e 1, e quanto mais alto o valor, maior o grau de arredondamento; e Rugosidade: que expressa as veias do agregado, com valores que variam entre 0 e 1, e quanto mais suave se aproxima de 1.

PECHE FILHO (2018), utiliza o método de avaliação visual, o Software ImageJ® e Lógica *Fuzzy*. Utiliza três indicadores para avaliar os agregados: a forma, a rugosidade de superfície e a presença de estruturas biogênicas, de acordo com sua variabilidade de forma, realiza uma classificação morfométrica determinando os valores do Diâmetro de Feret (Máximos e Mínimos) com o grau de desvio dos agregados em relação a um círculo perfeito, onde Diâmetro de Feret é o índice que expressa a forma do agregado no intervalo entre 0 e 1. O valor 1 está associado a valores idênticos nos diâmetros do agregado, assumindo a forma de um círculo perfeito (Redondo). O valor 0,5 representa uma diferença de 50% entre os diâmetros menor (Dmin) e maior (Dmax), assumindo a forma angular; e valores próximos a 0 diferenças máximas entre os diâmetros, assumindo a forma prismática.

SOUZA et al. (2018), avaliou a morfometria dos agregados, utilizando as características descritas em Silva et al. (2016c) e Carducci et al. (2016), que foram: Área (Ar): corresponde ao número de pixels no polígono; Perímetro (Per): comprimento da projeção do limite externo do agregado; Aspecto (Asp): que dá o resultado entre 0 e 1, e quanto maior o valor, maior o grau

de arredondamento; Arredondamento (Ard): medida dependendo da rugosidade da superfície externa do agregado. O resultado entre 0 e 1 em que, quanto maior o valor, maior o grau de arredondamento. Eles calcularam o Diâmetro Feret (DF), que é calculado na direção do comprimento do maior eixo do agregado e, portanto, contribui para a definição do aspecto do agregado e a Compacidade (CMP), que fornece uma medida da circularidade do objeto, e depende do comprimento do eixo mais longo. Varia de 0 a 1 e, se for igual a 1, o agregado é perfeitamente circular.

SILVA et al. (2016), indicam em suas pesquisas a utilização do software Quantporo, que tem a capacidade de determinar as dimensões geométricas e fatores de forma agregados. As variáveis a serem quantificadas para a análise da morfometria do agregado são: Aspecto (ASP): fornece o resultado entre 0 e 1, e quanto maior o valor, maior o grau de arredondamento, e a Rugosidade (RUG): expressa os veios do agregado, cujos valores variam de 0 a 1, com o mais plano próximo a 1 (quebra da borda). Também sustenta que o grau de arredondamento do agregado pode ser usado como um preditor da capacidade do agregado de transportar o solo através do Água.

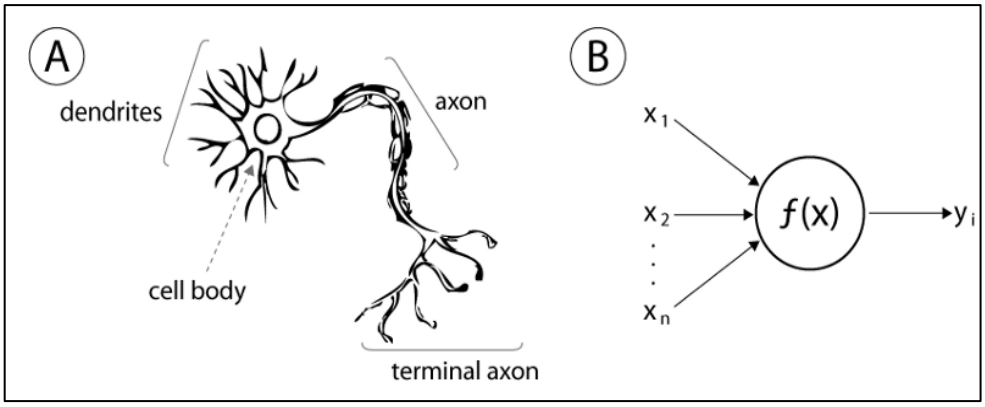
CLEMENTE & OLIVEIRA (2013), para a análise morfométrica, calcularam a porosidade e atributos morfométricos, tais como: Área, o número de pixels no polígono. Se a imagem estiver calibrada, a área será calculada na unidade de calibração; caso contrário, será em pixels. O Perímetro, o comprimento do lado externo do objeto. É uma medida fortemente influenciada pela resolução utilizada nos processos de exploração. Arredondamento, se o resultado fica entre 0 e 1. Quanto maior o valor, mais arredondado, se o valor for 1, o objeto é um círculo perfeito. Alongamento, é o quociente entre o eixo menor e o eixo maior. O resultado é um valor entre 0 e 1. Se for igual a 1, o objeto é aproximadamente circular ou quadrado. E a Compacidade: fornece a medida da circularidade do objeto. Seu valor está entre 0 e 1. Se for igual a 1, o objeto é aproximadamente circular. À medida que o valor se desvia de 1, o objeto torna-se menos circular.

3.2 Redes Neurais Artificiais3.2.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (em inglês, ANN, Artificial Neural Networks) são inspiradas nas redes neurais biológicas do cérebro humano. Eles são compostos de elementos que se comportam de maneira semelhante ao neurônio biológico (principalmente o que se refere aos

neurônios e suas conexões), sendo a unidade análoga ao neurônio biológico o elemento de processo (OLABE, 1998) (Figura 8A).

Figura 8 - Ilustrações lado a lado de neurônios biológicos e artificiais



Fonte: Hadican (2019).

As ANN são um campo muito importante dentro da Inteligência Artificial (IA), ele tenta criar modelos artificiais que resolvem problemas difíceis de resolver usando técnicas algorítmicas convencionais (Jorge, 2001; Andrade, 2013). O modelo de Rumelhart e McClelland (1986) define um Elemento de Processo (EP), ou neurônio artificial, como um dispositivo que, a partir de um conjunto de entradas, xi (i-1 ... n) ou vetor "x", gera uma única saída "y" (Figura 8B).

Há muitos precedentes históricos, no entanto, citaremos três aspectos que denotam o início das redes neurais e da computação propriamente dita. Em 1936, com Alan Turing foi iniciada a possibilidade de trabalhar com redes neurais de forma artificial, Alan foi o primeiro a estudar modelos analógicos ao cérebro humano dando-lhe uma forma computacional (ele encontrou semelhanças funcionais). Em 1943, Warren McChulloch e Walter Pitts foram os primeiros a explicar um neurônio com circuitos eletrônicos. E em 1957, Frank Rosenblatt, apresentou o Perceptron, uma rede neural com propagação para frente. Atualmente, existem inúmeras aplicações, publicações de trabalhos e avanços no campo das redes neuronais artificiais (SERNA, 2017).

3.2.2 Elementos básicos das redes neurais artificiais

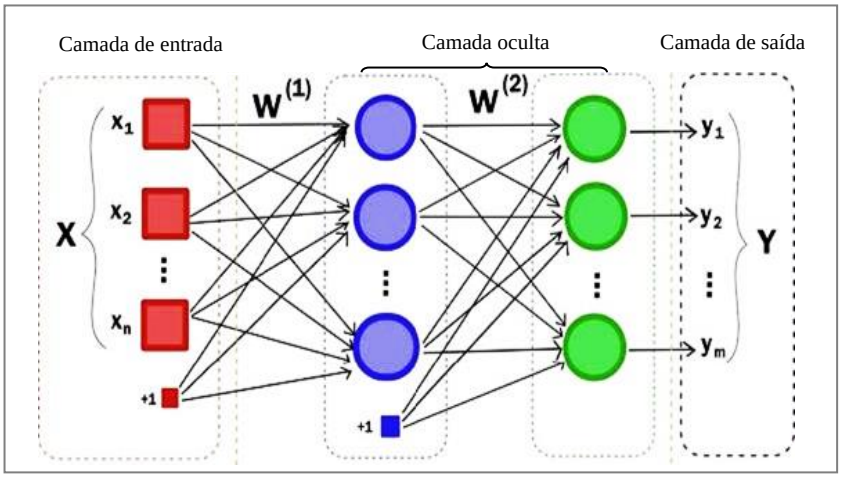
O funcionamento de uma rede neural artificial baseia-se em elementos que se comportam como um neurônio biológico em suas funções principais e um sistema cuja estrutura em camadas é semelhante à estrutura interconectada dos neurônios do cérebro humano. Um

neurônio é uma parte importante das redes neurais e pode haver muitos deles espalhados por várias camadas (LÓPEZ & FERNÁNDEZ, 2008, SERNA, 2017).

3.2.2.1 Camadas da rede neural artificial

As redes neurais artificiais podem ter qualquer número de camadas e qualquer número de nós por camada, as conexões são feitas entre neurônios de camadas diferentes, mas pode haver conexões entrelaçadas ou laterais e conexões de feedback que seguem uma direção oposta à de entrada-saída (GRAN JOSA, 2019). Observe-se a Figura 9 que existem três tipos de camadas em uma rede neural artificial:

Figura 9 - Arquitetura de uma rede neural artificial



Fonte: (Gran Josa, 2019)

a. Camada de entrada: Esta camada contém neurônios que representam os dados (X) que a rede neural usará para treinar. O número de neurônios nesta camada depende do número de características dos dados ou sinais do ambiente.

b. Camada oculta: Uma rede neural pode ter várias camadas deste tipo, não recebem nem fornecem informações ao ambiente (processamento interno da rede), cada camada contendo neurônios conectados a todos os neurônios da camada seguinte. Pela quantidade de camadas existe uma classificação de redes neurais por topologia e arquitetura, consideradas como Redes Neurais Monocamadas e Redes Neurais Multicamadas.

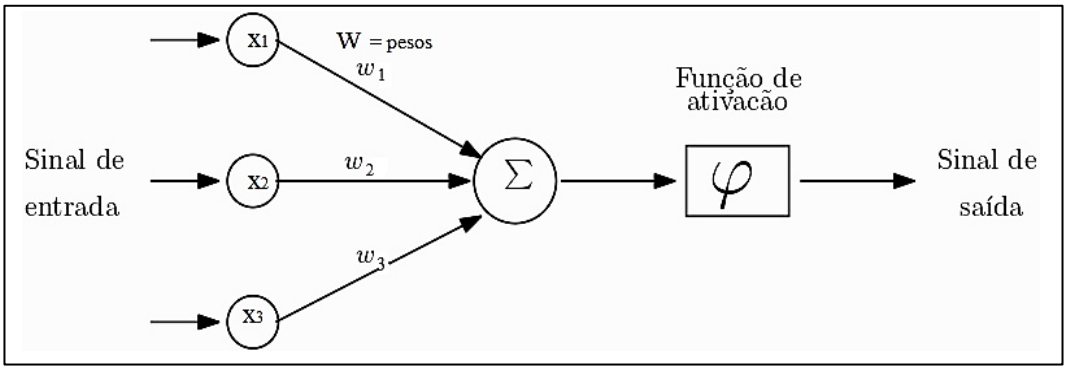
c. Camada de saída: Esta camada fornece a resposta da rede aos estímulos de entrada é responsável pela entrega dos resultados (Y). Em um problema de

classificação, esta camada terá um número de neurônios igual ao número de classes que existem nos dados. O resultado é uma lista de probabilidades para cada classe.

3.2.2.2 Pesos sinápticos

Os pesos são parâmetros importantes em uma rede neural, representam as informações (W) que serão utilizadas pela rede neural para resolver um determinado problema (Matich, 2001). Cada neurônio está conectado a outros neurônios, cada um dos quais com um peso associado. Os pesos são um valor numérico e pode ser alterado durante a fase de treinamento. Esses valores serão multiplicados com os dados de entrada X e com as saídas dos neurônios da camada anterior, cada valor W é diferente para cada neurônio como é mostrado na Figura 10 (SERNA, 2017; RODRIGUEZ, 2018).

Figura 10 - Modelo de uma red neural

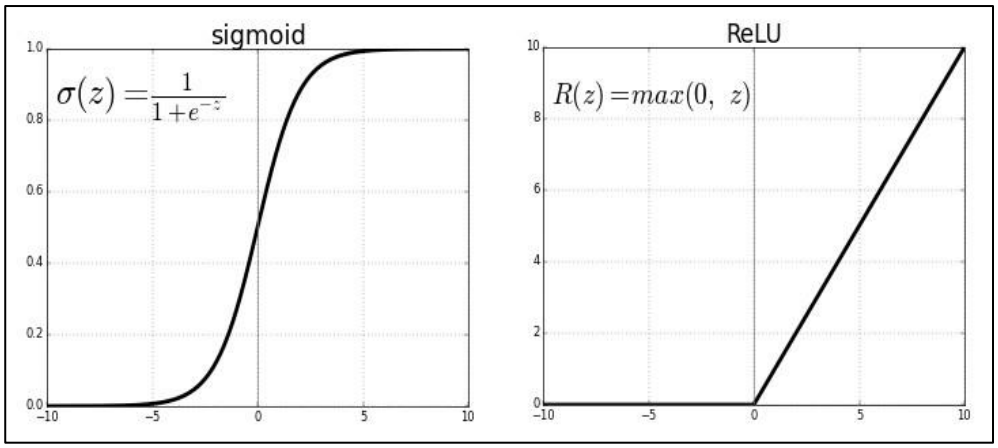


Fonte: adaptado de (GRUBLER, 2018)

3.2.2.3 Função de ativação

A função de ativação ou transferência tem como objetivo limitar a amplitude de saída do neurônio, dentro de um intervalo fechado, como [0,1], podendo ser interpretado também como a probabilidade do resultado. Existem diversas funções de ativação lineares como a função degrau e identidade e não lineares como a gaussiana, tangente hiperbólica, sigmoide, entre outras, que testam o potencial de ativação (GRUBLER, 2018). As funções de ativação mais utilizadas são sigmoide e Relu, os gráficos são mostrados na Figura 11:

Figura 11 - As funções de ativação mais utilizadas são sigmoide e Relu



Fonte: (RODRIGUEZ, 2018)

3.2.3 Parâmetros básicos das redes neurais artificiais

3.2.3.1 Época (*Epoch*)

Este é o número de vezes que os algoritmos serão executados. Em cada ciclo (época) todos os dados de treinamento passam pela rede neural para que ela aprenda sobre eles, se houver 10 ciclos e 1000 dados, a cada ciclo os 1000 dados passarão pela rede neural (RODRIGUEZ, 2018).

3.2.3.2 Tamanho dos dados (*Batch size*)

É o número de dados que cada iteração de um ciclo (época) possui, isso é útil porque a rede neural atualiza os parâmetros. Para grandes quantidades de dados precisam-se de computadores com mais memória, a maior quantidade dados cada ciclo demora mais para ser executado (RODRIGUEZ, 2018).

3.2.3.3 Taxa de aprendizagem (*Learning rate*)

Este hiper parâmetro que define a rapidez com que a rede neural atualiza os conceitos que aprendeu. Taxas de aprendizado menores requerem mais períodos de treinamento (requerem mais tempo para treinar). Por outro lado, se os valores forem muito altos, a atualização pode ir além do ponto ideal de aprendizado e requerem menos períodos de treinamento (SMITH, 2018).

3.2.3.4 Função perda (*Lost function*)

A função perda, também conhecida como função custo, é a função que nos mostra quão bom é o desempenho da rede neural, um resultado alto indica que a rede neural tem um

desempenho ruim e um resultado baixo indica que a rede neural está aprendendo satisfatoriamente (RODRIGUEZ, 2018).

3.2.4 Fases de uma rede neural artificial

Há duas fases em todas as aplicações de redes neurais: a fase de aprendizado ou estágio e a fase de teste.

3.2.4.1 Fase de aprendizagem ou treinamento:

O processo de aprendizagem é através da experiência e pela extração de conhecimento genérico de um conjunto de dados (LÓPEZ & FERNÁNDEZ, 2008), uma rede neural artificial pode ser treinada para reconhecer padrões, classificar dados e prever eventos futuros (MATLAB, 2020).

Uma das principais características das redes neurais artificiais é sua capacidade de aprender (OLABE, 1998) As redes neurais têm a capacidade de aprender e melhorar seu funcionamento, semelhante ao dos seres humanos. Aprender significa que problemas que inicialmente não podiam ou não podem ser resolvidos, podem ser resolvidos após aprender mais sobre o assunto (MATICH, 2001).

O aprendizado é um processo de ajuste dos valores de peso entre as conexões das camadas. Os pesos são adaptados de acordo com as informações extraídas das novas normas de treinamento apresentadas (GUTIÉRREZ, 2000). O processo de aprendizagem pode ser dividido em três grandes grupos, de acordo com suas características:

a. Aprendizagem supervisionada. A rede neural é dotada de um conjunto de padrões de entrada, juntamente com a saída esperada, e aprende por associação. Os pesos são modificados em proporção ao erro que ocorre entre a saída real da rede e a saída esperada (GUTIÉRREZ, 2000; LARA, 2000).

b. Aprendizagem sem supervisão. A rede neural não requer um tutor externo. Os dados são simplesmente apresentados à rede, que, de acordo com eles, cria clusters internos que comprimem os dados de entrada em um certo número de categorias de classificação (LARA, 2000).

c. Aprendizagem com o Reforço: É baseado no fato de que não temos um modelo completo do comportamento que queremos. O supervisor é reduzido para indicar por meio de um sinal de reforço se a saída obtida na rede está ajustada à desejada. Se for bem sucedido é 1, caso contrário -1. Os pesos serão baseados em um mecanismo de probabilidade (BAKER et al., 2017).

3.2.4.2 Fase de teste:

Os resultados podem ser calculados ao mesmo tempo e adaptados iterativamente, de acordo com o tipo de rede neural, e com base em equações de teste dinâmicas. Uma vez calculados os pesos da rede, os neurônios da última camada são comparados com a saída desejada para determinar a validade do projeto (GUTIERREZ, 2000).

Em geral, as redes neurais artificiais consistem em unidades de processamento que trocam dados ou informações, são utilizadas para reconhecer padrões, incluindo imagens, manuscritos e sequências de tempo (FIGUEREDO & SALINAS, 2015). Existe uma grande variedade de tipos ou modelos de redes neurais artificiais de acordo com sua arquitetura e propósito.

3.2.5 RNA utilizadas em trabalhos de classificação e estudo de solos

Os tipos de redes neurais podem ser descritos com base na maneira como os neurônios são organizados, como eles aprendem ou o número de camadas. Além disso, dependendo do objetivo e dos recursos disponíveis para a investigação, algumas redes neuronais serão mais aplicáveis do que outras.

TRUEVA et al. (2004), realizaram a identificação de áreas erodidas ao tratar imagens digitais com uma rede neural, propuseram uma metodologia para identificar a erosão da água através da análise de imagens digitais de satélites. Como resultado, eles conseguiram que a rede pudesse identificar pixels representando erosão em rochas brancas, com um erro de 2,5%; transição de branco para amarelo com 16%, árvores com 13,5% e solos cobertos com culturas arvenses (cultivos) com 7,1%.

Chagas *et al*. (2013) utilizaram dois diferentes classificadores (redes neurais artificiais e um algoritmo de máxima verossimilhança) na predição de classes de solos. Foram usados como variáveis discriminantes atributos do terreno, elevação, declive, aspecto, curvatura plana e

índice topográfico composto (CTI) e índices de minerais de argila, óxido de ferro e Índice de Vegetação por Diferença Normalizada (NDVI), derivados de imagens do sensor Landsat 7 ETM +. Os dois classificadores foram treinados e validados para cada classe de solo. Os resultados, de acordo com os testes estatísticos, a precisão do classificador baseado em redes neurais artificiais (ANNs) foi maior do que o classificador de máxima verossimilhança (MLC) clássico. A comparação dos resultados com os pontos de referência mostrou que o mapa de RNA resultante (73,81%) foi superior ao mapa MLC (57,94%).

Nesse viés, Ribeiro *et al* (2016) utilizaram um modelo de rede neural tipo mapa de Kohonen para classificar diferentes grupos de solos degradados em minas a céu aberto na floresta Amazônica com a finalidade de recupera-los, os resultados mostraram quatro grupos distintos e identificaram uma relação entre as diferentes texturas dos solos em função do método de recuperação aplicado.

Torkashvand & Author, (2017), realizaram a estimativa de índices de estabilidade de agregados do solo usando rede neural artificial e modelos de regressão linear múltipla, foram avaliadas as capacidades do modelo de regressão linear múltipla (MLR) e das redes neuronais artificiais (ANNs) para estimar o diâmetro ponderado médio (MWD) a partir das propriedades clássicas do solo e da combinação destas propriedades do solo e da dimensão fractal dos agregados. Os resultados da pesquisa mostraram que o modelo ANN para estimar o MWD é mais exato que o modelo MLR, com uma *accuracy* de 87% e 77% respectivamente.

Padarian, Minasnya e Mcbratneya (2018), utilizaram aprendizagem profunda e espectroscopia do solo para prever as propriedades do solo a partir uma pilha 3-D de imagens de espectros de solo bruto, desenvolvendo um projeto de rede neural convolucional (CNN). Para este estudo específico tentou se prever o conteúdo de carbono orgânico, capacidade de troca cármica, fração granulométrica de argila de areia. O estudo se concentrou no potencial de aprendizagem multitarefa para produzir algum tipo de efeito sinérgico. Os pesquisadores indicam que uma CNN pode lidar com um grande número de covariáveis e tem vantagens sobre outros algoritmos de aprendizado de máquina usados no mapeamento digital de solo (DSM).

Em outra pesquisa com a utilização de dados numéricos, CHITERO *et al*., (2020), utilizaram uma ANN de tipo Perceptron Multicamadas (MLP), para estimar os níveis de

recuperação do solo, recuperado, parcialmente recuperado e não recuperado em função de atributos físicos. Foram avaliadas as interações entre os indicadores: grau de saturação, teor de umidade, índice de vazios, porosidade e o DMG dos Agregados. Obtiveram resultados com um erro quadrado médio baixo, concluindo que a rede neural treinada pode ser uma alternativa interessante e automática para a classificação e análise de solos em recuperação.

Schlüter *et al.*, (2020) avaliaram a relação entre a estrutura do solo e as funções do solo por meio de imagens em escala de poros (tomografia de raios-X). Parte do desenvolvimento da pesquisa, os autores indicam que os métodos de avaliação do solo mediante imagem, sofrem de limitações de acesso adequado ao hardware necessário e de algum grau de subjetividade nos protocolos de processamento de imagem. A segmentação da imagem e extração de características seja provocada pela aprendizagem mecânica e por técnicas de aprendizagem profunda requer algum nível de informação do utilizador em termos de dados de formação.

Azizi et al (2020), conseguiram uma *accuracy* média de classificação acima de 95% sendo a maior *accuracy* alcançada com a arquitetura ResNet50 (98,72%), em um estudo desenvolvido para treinar as arquiteturas de redes neurais convolucionais, VggNet16, ResNet50 e Inception-v4 na classificação de tamanhos de agregados dos solos por meio de imagens estéreo. Uma rede neural convolucional (CNN) foi usada por causa de seus sucessos significativos no processamento de imagem. Além disso, transferir aprendizagem de uma área do conhecimento que tem acesso a dados abundantes para resolver problemas com menos dados. Transferência de aprendizagem, uma técnica utilizada devido à não necessidade de fornecer conjuntos de dados do tamanho do ImageNet e à inacessibilidade a dados tão massivos.

Muitos trabalhos sobre redes neurais se aplicam no diagnóstico ou tratamento de determinados problemas. Nas ciências médicas existem vários modelos de ANN para o diagnóstico de tumores a partir de imagens, principalmente um tipo de rede neural chamada Redes Neurais Convolucionais (CNN) (PEREIRA *et al* 2016); (TORREGROSA LLORET, 2018); (JOSÉ & ORTEGA, 2019). As ANN também são utilizadas na classificação de frutas, na detecção de doenças em frutas, recentemente é proposto o método de classificação baseado em Rede Neural de Convolução (CNN) que dá um melhor resultado de classificação de 90% em comparação com outras metodologias propostas até agora (YAMPARALA *et al*., 2020).

Dessa forma, entende-se que as CNN possuem múltiplos usos e muitas potencialidades de aplicação na classificação de imagens nas diferentes áreas do conhecimento humano.

Na presente pesquisa para avaliar e classificar a forma dos agregados a partir de imagens digitais, foi utilizada as Redes Neurais Convolucionais, uma das redes neurais mais desenvolvidas atualmente no campo da classificação de imagens, por serem as amplamente adaptadas para classificação de imagens com bom desempenho (QUINTERO et al., 2018, Azizi *et al*., 2020). Nesse escopo, aprofundaremos um pouco mais neste tipo de rede neural.

3.2.6 Redes Neurais Convolucionais (CNN)

As Redes Neurais Convolucionais (em inglês ConvNet / Convolutional Neural Network / CNN) um tipo de rede de inspiração biológica que simula a forma como a visão humana funciona, é uma modificação da percepção multicamadas, especialmente projetada para lidar com dados de entrada bidimensionais (SHARMA et al., 2018).

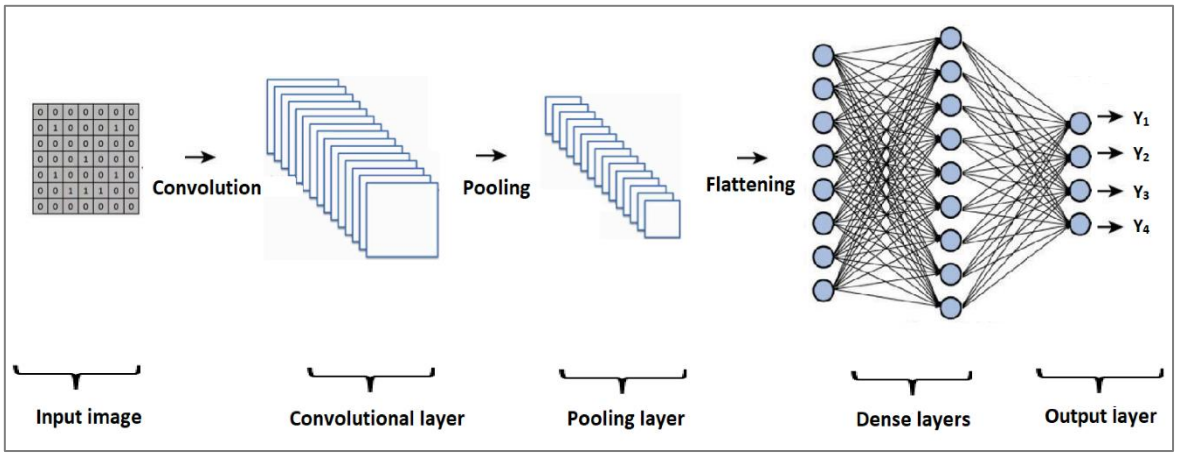
O nome "rede neural de convolução" indica que a rede emprega uma operação matemática chamada convolução em vez da multiplicação geral da matriz, em pelo menos uma de suas camadas (TRAORE et al., 2018, DHILLON & VERMA, 2020). A CNN é uma classe de redes neurais artificiais profundas (*Deep Learning*) que exploram a correlação espacial local aplicando um padrão de conectividade local entre neurônios em camadas adjacentes (MAEDA-GUTIÉRREZ et al., 2020). Da mesma forma, tem pelo menos uma camada convolucional e está composta de neurônios, em que cada neurônio tem um peso e um viés que podem ser aprendidos (GUO et al., 2017; DHILLON & VERMA, 2020).

Uma vantagem da CNN é que mantem as características espaciais de uma imagem, como a altura e largura, e cores. Também, é importante indicar que quando as imagens são interpretadas por um computador, nada mais são do que valores numéricos armazenados em uma matriz ou tensor. Assim, dentre os diferentes tipos de modelos de redes neurais artificiais, as redes neurais convolucionais têm demonstrado alto desempenho na classificação de imagens (SUDEEP & PAL, 2017; GUO et al., 2017).

A arquitetura de uma CNN é composta por vários planos bidimensionais, e cada plano consiste em vários neurônios independentes (SHARMA et al., 2018). A Figura 12 ilustra a arquitetura geral de uma CNN, com seus principais elementos, como camada de entrada (*Input*

image), camada de saída (*Output layer*) e várias camadas ocultas, onde a camada oculta consiste principalmente em três tipos, a camada convolucional (*Convolutional layer*), camada de pooling (*Pooling layer*) e camada totalmente conectada (*Dense layers*) a (GUO et al., 2017; TRAORE et al., 2018; DHILLON & VERMA, 2020).

Figura 12 - Representação da arquitetura de uma rede neural convolucional (CNN)



Fonte: (MAEDA-GUTIÉRREZ et al., 2020)

Na CNN, a camada convolucional, aceita dados da camada de entrada e realiza o aprendizado de múltiplos filtros, onde cada filtro - ou *kernel* - extrai uma informação da imagem (JAN et al., 2019). Para o desenvolvimento de uma CNN, o módulo Keras fornece uma camada convolucional para imagens, chamada de Conv2D, e outras camadas (ou *layers*) para se utilizar em combinação sequencial, como *MaxPooling2D, Dropout, Flatten, Dense e Activation.* A camada de pooling é usada para reduzir a dimensionalidade, associando a saída do cluster de neurônios em uma camada com o neurônio único. E a camada totalmente conectada tem como principal função classificar as imagens de entrada em várias classes, com base nos conjuntos de dados de treinamento (DHILLON & VERMA, 2020). A Tabela 1 apresenta as diferentes camadas de uma arquitetura geral de CNN.

Tabela 1 - Os parâmetros de diferentes camadas de uma arquitetura CNN

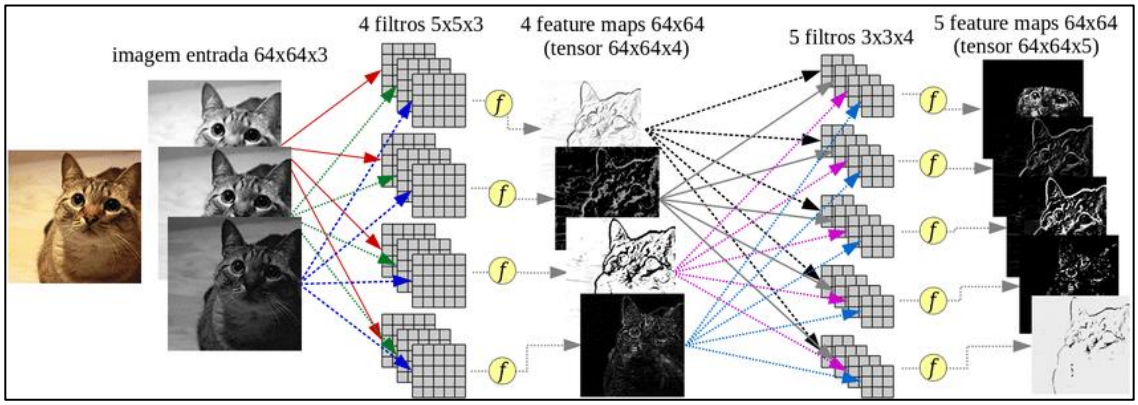
Camadas	**Descrições de parâmetros**
Entrada (Input)	Arquivos de imagem (jpeg, gif, ...) descritos com um formato RGB que pode ser modelado como uma matriz 3D.
Conv (x, y, z, u, v, h, i, j)	x = número de canais de entrada esperados na imagem dada (3 para RGB) y = número de canais de saída após a fase de convolução (CONV) z = A largura do kernel da convolução u = A altura do kernel da convolução

	v = O passo (passo) da convolução na dimensão da largura h = O passo (passo) da convolução na dimensão da altura. i = Zeros adicionais adicionados aos dados do plano de entrada em ambos os lados do eixo de largura j = Zeros adicionais adicionados aos dados do plano de entrada em ambos os lados do eixo de altura
ReLU (x)	Uma unidade linear retificada (função de ativação) tem saída 0 se a entrada for menor que 0, e saída bruta caso contrário. x = verdadeiro ou falso.
MaxPooling (x, y, z, u)	Uma operação de pooling máximo que examina janelas XxY e encontra o máximo por comprimento de passada ZxU. x = A largura do filtro do pooling y = A altura do filtro do pooling z = O passo da largura do pool u = O stride da altura do pool
FullyConnected (x, y)	x = Tamanho da imagem de entrada, por exemplo, 3 * 256 * 256 (3 canais de cores, 256 pixels de altura e 256 pixels de largura) y = Número de classes de saída menor que o tamanho da imagem de entrada
Perda (*Loss)* (x, y)	A função de perda leva os rótulos previstos e verdadeiros e calcula um valor que quantifica o desempenho do modelo. x = o rótulo previsto y = rótulos verdadeiros
Resultado (*Output*)	Análise e classificação de padrões (por exemplo, reconhecimento de imagem, análise de vídeo, processamento de linguagem natural, etc.)

Fonte: (Traore et al., 2018)

A figura 13 ilustra uma CNN com dois camadas de convolução, uma primeira camada convolucional com 4 filtros 5 × 5 × 3, que recebe como entrada uma imagem RGB 64 × 64 × 3, e produz um tensor com 4 *feature maps* (processo de extração de características da imagem); a segunda camada convolucional contém 5 filtros 3 × 3 × 4 que filtram o tensor da camada anterior, produzindo um novo tensor de *feature maps* com tamanho 64 × 64 × 5. Na convolução o *feature maps* é principalmente a aplicação de filtros na imagem de entrada, assim extrair as características da imagem (PONTI & DA COSTA, 2018).

Figura 13 - Exemplo da camada de convolução de uma CNN



Fonte: (PONTI & DA COSTA, 2018)

Pelo rápido desenvolvimento das CNNs, muitas arquiteturas poderosas surgiram, como o modelo clássico LeNet para AlexNet, ZFNet, GoogleNet, VGGNet, MobileNet, ResNet, SENet, DenseNet, etc, umas mais desenvolvidas do que outras, além da complexidade, e o número de parâmetros utilizados (MAEDA-GUTIÉRREZ et al., 2020; DHILLON & VERMA, 2020). Na Tabela 2 se mostra os modelos de CNN que Keras disponibiliza de aprendizado profundo com pesos pré-treinados. Os modelos podem ser usados para previsão, extração de recursos e ajuste. A precisão se refere ao desempenho do modelo no conjunto de dados de validação do ImageNet e a profundidade se refere à profundidade topológica da rede.

Tabela 2 - Modelos disponíveis de CNN

Modelo	**Tamanho**	**Precisão principal**	**5 principais precisão**	**Parâmetros**	**Profundidade**
Xception	88 MB	0,790	0,945	22.910.480	126
VGG16	528 MB	0,713	0,901	138.357.544	23
VGG19	549 MB	0,713	0,900	143.667.240	26
ResNet50	98 MB	0,749	0,921	25.636.712	-

ResNet101	171 MB	0,764	0,928	44.707.176	-
ResNet152	232 MB	0,766	0,931	60.419.944	-
ResNet50V2	98 MB	0,760	0,930	25.613.800	-
ResNet101V2	171 MB	0,772	0,938	44.675.560	-
ResNet152V2	232 MB	0,780	0,942	60.380.648	-
InceptionV3	92 MB	0,779	0,937	23.851.784	159
InceptionResNetV2	215 MB	0,803	0,953	55.873.736	572
MobileNet	16 MB	0,704	0,895	4.253.864	88
MobileNetV2	14 MB	0,713	0,901	3.538.984	88
DenseNet121	33 MB	0,750	0,923	8.062.504	121
DenseNet169	57 MB	0,762	0,932	14.307.880	169
DenseNet201	80 MB	0,773	0,936	20.242.984	201
NASNetMobile	23 MB	0,744	0,919	5.326.716	-
NASNetLarge	343 MB	0,825	0,960	88.949.818	-
EfficientNetB0	29 MB	-	-	5.330.571	-
EfficientNetB1	31 MB	-	-	7.856.239	-
EfficientNetB2	36 MB	-	-	9.177.569	-
EfficientNetB3	48 MB	-	-	12.320.535	-
EfficientNetB4	75 MB	-	-	19.466.823	-
EfficientNetB5	118 MB	-	-	30.562.527	-
EfficientNetB6	166 MB	-	-	43.265.143	-
EfficientNetB7	256 MB	-	-	66.658.687	-

Fonte: (Keras, 2021)

4. MATERIAIS E MÉTODOS

Nessa seção do trabalho são descritas as características dos equipamentos, softwares e procedimentos desenvolvidos no projeto.

4.1 Materiais utilizados

Para o desenvolvimento das etapas da metodologia foram utilizados um notebook, com sistema Operacional Windows 10 (Figura 14).

Figura 14 - Especificações do notebook

Nome do dispositivo	LAPTOP-HL690BTI
Processador	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
RAM instalada	8,00 GB (utilizável: 7,89 GB)
ID do dispositivo	4A477A6A-B8F5-4E0B-8272-9B8889B59DC9
ID do Produto	00327-30719-27016-AAOEM
Tipo de sistema	Sistema operacional de 64 bits, processador baseado em x64

Fonte: Autoria própria (2020).

4.2 Geração de *data_set*

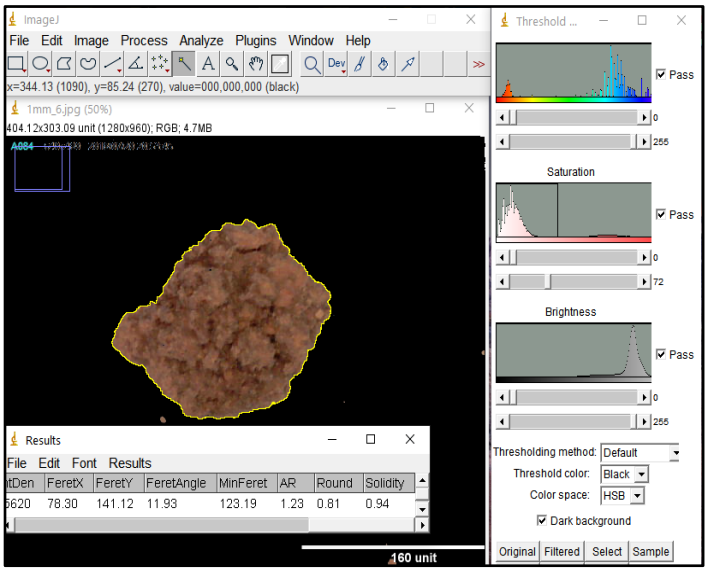
4.2.1 Banco de dados (Imagens)

As imagens digitais dos agregados do solo foram disponibilizadas através da base de dados da análise visual realizada por PECHE FILHO (2018). Os agregados são de origem agrícolas e foram fotografados utilizando um microscópio digital, Dino Lite, modelo AM211.

4.2.2 Levantamento de propriedades e classificação

Nesta etapa, as imagens foram padronizadas, organizadas e seguidamente avaliadas com o software ImageJ para obtenção de dados quantitativos, como diâmetro Feret, arredondamento, área e circularidade (Figura 15). Considerando por exemplo o arredondamento, em intervalos equidistantes, e atribuindo um rango de valores para as classes de forma (0 a 0,2 - prismático; 0,2 a 0,4 -angular, 0,4 a 0,6 – sub angular, 0,6 a 0,8 - arredondado, 0,8 a 1,0 - redondo).

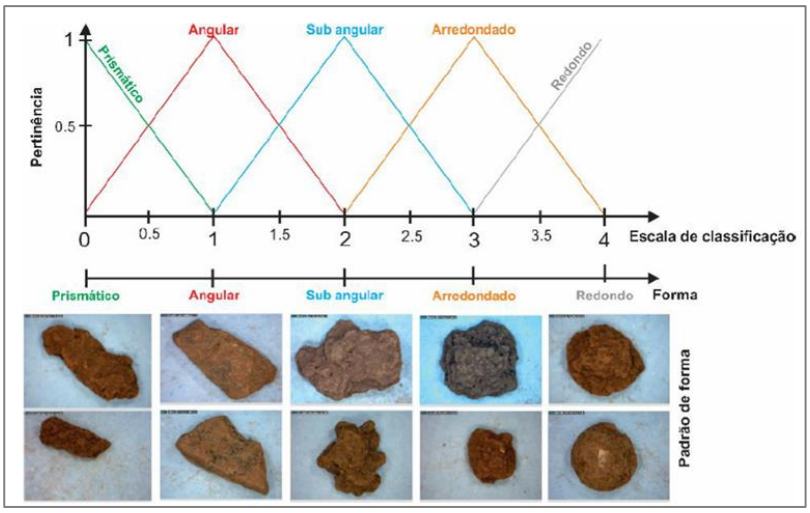
Figura 15 - Exemplo do processamento das imagens no software ImageJ



Fonte: Autoria própria (2020).

Também foi empregado o método de avaliação visual, segundo a padrão de forma proposto por PECHE FILHO (2018), prismático; angular, sub angular, arredondado e redondo (Figura 16).

Figura 16 - Classificação dos agregados de acordo com a forma



Fonte: (PECHE FILHO, 2018)

4.2.3 Pré processamento de dados

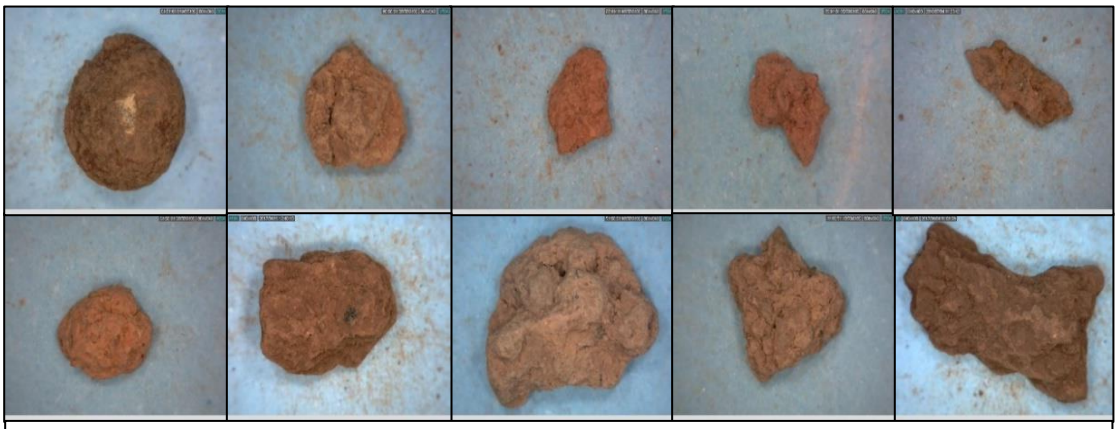
Verificou-se as dimensões das imagens originais, e o resultado dessa verificação foi: 2960 imagens com a dimensão 960 x 1280 x 3 e 1390 imagens com a dimensão 480 x 640 x 3, seguidamente se fez o redimensionamento das imagens (Anexo 3) para o tamanho 480 x 480 x 3, assim padronizando todas as imagens da base de dados e a segmentação (para excluir o plano de fundo).

Para acrescentar a base de dados do conjunto de treinamento, foi realizado o procedimento de *data_augmentation* (Anexo 4), espelhamento das imagens originais e rotacionadas (90, 180 e 270 graus). Com o procedimento, a partir de 4350 imagens únicas foram geradas 30450 imagens com dimensões de 480 x 480 x 3.

Posteriormente continuou-se com a criação da *data_set*, dividindo as amostras em 80% para o conjunto de treinamento e 20% para o conjunto de validação e o conjunto de teste.

a) Experimento 1: O conjunto de dados possui 5 classes que são ilustrados na Figura 17, com os seguintes valores: redondo (1980) /, arredondado (8500), subangular (9200) /, angular (7902) /, prismático (2868) /. Em um total de 30450 imagens.

Figura 17 - Classificação dos agregados em 5 classes

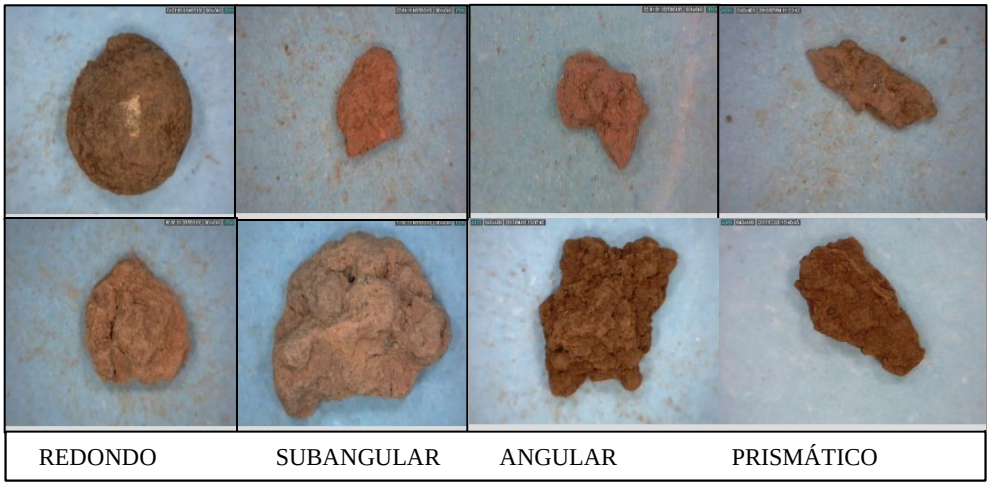


Fonte: Autoria própria (2020).

b) Experimento 2: No primero experimento, a maioria dos conjuntos de dados de classificação não tiveram exatamente o mesmo número de instâncias em cada classe, existe um método de balanceamento chamado *Under sampling, consiste em* fazer a redução da quantidade de imagens da classe majoritária de forma que o número de imagens se torne igual ao da classe minoritária (Yen & Lee, 2006).

   A fim de melhorar o desempenho da rede, depois, de avaliar o modelo do Experimento 1, optou-se por realizar *Under sampling* e alguns ajustes na classificação com o método de análise visual e *data cleaning*, refere-se limpeza de dados à identificação e correção de erros no conjunto de dados que podem impactar negativamente um modelo. Assim o conjunto de dados do Experimento 2, possui 4 classes (Figura 18) que foram balanceadas, com os seguintes valores: redondo (2500) /, subangular (2500) /, angular (2500) /, prismático (2500) /. Com um total de 10000 imagens.

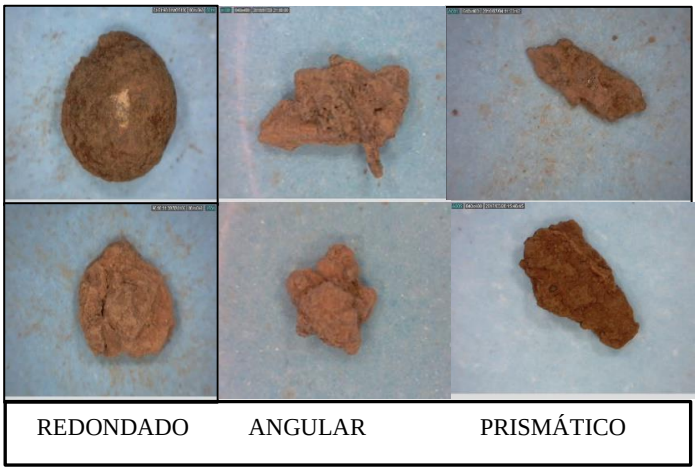
Figura 18 - Classificação dos agregados em 4 classes



Fonte: Autoria própria (2020).

c) Experimento 3: depois do balanceamento feito no Experimento 2, no Experimento 3 se organizou a *data_set* em classes em uma condição de extremos (Redondado, Angular e Prismatica). Assim, o conjunto de dados conta 3 classes que são ilustrados na Figura 19, com os seguintes valores: redondado (2500) /, angular (2500) /, e prismático (2500) /. Em um total e 7500 imagens.

Figura 19 - Classificação dos agregados em 3 classes



Fonte: Autoria própria (2020).

4.3 Desenvolvimento

Nessa seção, são descritos os programas utilizados para o desenvolvimento da Rede Neural.

a) Linguagem de programação Python

Para o desenvolvimento do trabalho foi utilizado a linguagem Python de código aberto. Em comparação com outras linguagens de programação, Python possui uma maior variedade de módulos padrões que são muito úteis, dependendo do objetivo. As características positivas para a utilização da linguagem Python são sua simplicidade, precisão na síntese, boa legibilidade e boa funcionalidade para algoritmos complexos. Essa linguagem possui uma biblioteca padrão que oferece suporte a muitas tarefas de programação comuns, como conexão a servidores web, pesquisa de texto com expressões regulares, leitura e modificação de arquivos (PYTHON SOFTWARE FOUNDATION, 2020).

Para a organização dos arquivos e desenvolvimento do projeto (dados coletados, arquivos de texto e programação algoritmos em Python utilizou-se a IDE *Spyder* (SPYDER, 2019). A Figura 7 mostra a IDE do algoritmo em Python. A versão de Python utilizada para o projeto foi a 3.7 e utilizou-se também a plataforma de distribuição Anaconda para instalação e manutenção das bibliotecas. As bibliotecas utilizadas foram TensorFlow e Keras.

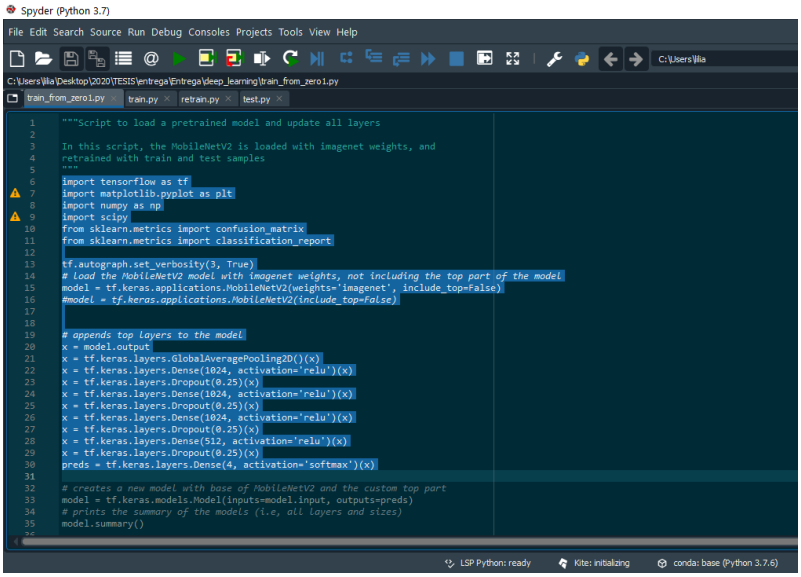
b) TensorFlow e Keras

TensorFlow e Keras são uma biblioteca de código aberto criada para aprendizado de máquina, uma biblioteca escrita na linguagem Python. O TensorFlow pode treinar e executar redes neurais profundas para classificação manuscrita de dígitos, reconhecimento e classificação de imagens, podemos utilizar modelos, detecção de objetos, detecção de textos e reconhecimento de comandos por voz (TensorFlow, 2020).

Keras fornece abstrações essenciais e blocos de construção para desenvolver e enviar soluções de aprendizado de máquina com alta velocidade de interação. O TensorFlow possui funções prontas de estruturas de redes neurais dos mais variados tipos (redes convolucionais, redes neurais recorrentes funções de ativação, otimizadores etc.), com isso não é necessário criar um código do zero. Então, pelas características mencionadas para o desenvolvimento da rede neural utilizou-se o TensorFlow e Keras.

Keras vem empacotado com TensorFlow 2.0 como "*tensorflow. keras*". Para começar a usar o Keras, se realizou a instalação do TensorFlow 2.0 que é compatível com Python 3.7 e Windows 10 (Figura 20).

Figura 20 - IDE Spyder e algoritmo em Python3 com as bibliotecas



Fonte: Autoria própria (2020).

4.4 Descrição da arquitetura

Nessa seção, são descritos os funcionamentos da arquitetura desenvolvida nesse projeto.

MobileNet: MobileNet é uma classe da CNN de código aberto pelo Google, supera o desempenho do GoogleNet e do VGGNet (DHILLON & VERMA, 2020). É um excelente ponto de partida para treinar uma rede classificadora, portanto, isso facilitou a eleição do modelo para o projeto (Figura 21).

Figura 21 - Modelo de CNN desenvolvida

```
Deep learning:
        Modelo:
                MobileNetV2 (excluindo top layer)
                Top layer:
                        Dense(1024, ReLU)
                        Dropout(0.5)
                        Dense(1024, ReLU)
                        Dropout(0.5)
                        Dense(1024, ReLU)
                        Dropout(0.5)
                        Dense(512, ReLU)
                        Dropout(0.5)
                        Dense(5, softmax) <-- Final
                Loss function:
                        categorical_crossentropy
                Optimizer:
                        Adam
                Learning rate:
                        0.001
                Epochs:
                        20
                Transfer Learning:
                        MobileNetV2 com pesos pré-treinados no data set imagenet
```

Fonte: Autoria própria (2020).

MobileNet é uma simplificação de redes neurais para possibilitar o seu uso em aplicações web, com isso é possível criar rapidamente uma aplicação de reconhecimento de imagens. MobileNets é uma classe de convolução de redes neurais projetadas por pesquisadores do Google, baseados em uma arquitetura simplificada que usa convoluções separáveis em profundidade para construir redes neurais profundas leves (HOWARD et al., 2017). Na convolução as imagens em pixels são convertidas em dados numéricos. Esse processo é repetido para cada bloco por várias redes neurais até chegar um grupo de números que definem a imagem.

Por padrão, MobileNet utiliza o dataset ImageNet com mais de 1.500.000 de imagens divididas em 1.000 categorias. O Google fez o código fonte MobileNet aberto com 16 pontos de verificação, com isso é possível treinar modelos próprios reutilizando a estrutura da mobileNet. O modelo MobileNet é o primeiro modelo de visão de computador móvel do TensorFlow (PUJARA, 2020).

O modelo MobileNetV2 está baseado em convoluções separáveis em profundidade, que é uma forma de convoluções fatoradas que fatoram uma convolução padrão em uma convolução em profundidade e uma convolução 1 × 1 chamada de convolução pontual. Todas as camadas são seguidas por uma *batchnorm* (normalização da camada de entrada, centralizando e

redimensionando) e não linearidade ReLU, com exceção da camada final camada softmax para classificação. A Figura 22 e mostra a arquitetura, o MobileNetV2 em total tem 28 camadas (HOWARD et al., 2017).

Figura 22 - Arquitetura da MobileNet2

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fonte:(HOWARD et al., 2017)

O principal objetivo do uso de modelos CNN pré-treinados está relacionado ao treinamento rápido e fácil usando pesos inicializados aleatoriamente, bem como a obtenção de erros de treinamento menores do que uma rede neural artificial não pré-treinados, isto é possível por transferência de aprendizagem ("*Transfer learning" em inglês*).

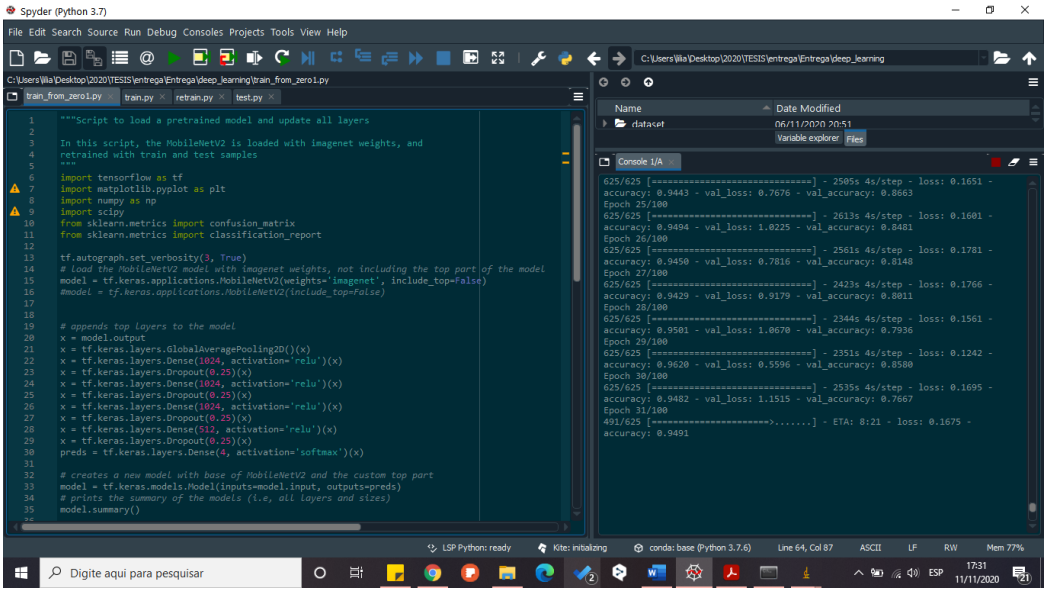
4.5 Treinamento e teste da rede neural

Após a aquisição das imagens e ter a *data_set*, 80% das imagens para o conjunto de treinamento e 20% para o conjunto de validação e 100 imagens para o conjunto do teste, utilizou-se o algoritmo "*train_from_zero1.py*" para o treinamento e validação (Anexo 1), e

utilizou-se o algoritmo "*Test.py*" para obter os resultados de precisão da rede (Anexo 2). O treinamento e teste da rede neural foi o qual dedicou-se a maior parte do tempo no decorrer do projeto, devido a avaliação do modelo utilizado durante o processo.

O algoritmo da rede neural faz uso de "*tensorflow-keras"* para treinar as imagens no modelo de aprendizagem profunda, *"matplotlib.pyplot*" para criação de gráficos e visualizações de dados, "*numpy*" para processamento numérico, e "*sklearn*" para imprimir um relatório de classificação (Figura 23).

Figura 23 - IDE do treinamento da Rede Neural

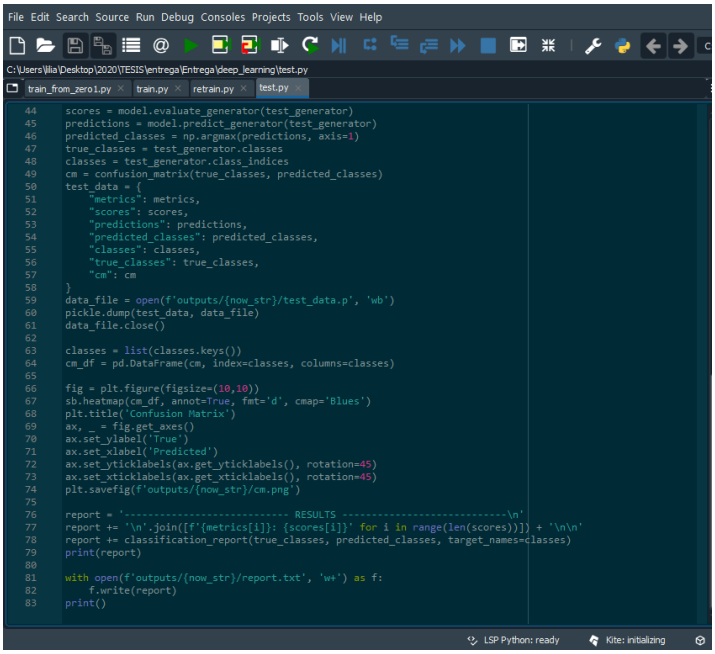


Fonte: Autoria própria (2020).

Em geral, para todos os treinamentos realizados durante o projeto, foram trabalhados com um número médio de 20 *"epochs"* (épocas) e 32 "*batch_size"* (tamanhos de lotes). O "*batch_size"* 32 é adequado para a maioria dos sistemas (CPUs). É importante mencionar que o treinamento do modelo levou muito tempo (cerca de quinze horas no computador CPU). Entretanto, se uma Unidade de Processamento Gráfico (GPU) tivesse sido utilizada, o valor de *"epochs"* e "*batch_size"* poderia ser aumentado e o tempo de treinamento houvesse sido muito mais rápida. Uma GPU é uma unidade de processamento gráfico que acelera o desempenho de aplicações de aprendizagem profunda (NVIDIA, 2020).

Após selecionar, implementar e treinar um modelo de rede neural, o próximo passo foi avaliar quão eficaz o modelo, baseado em algumas métricas usando conjuntos de dados de validação (Figura 24).

Figura 24 - IDE do algoritmo do teste



Fonte: Autoria própria (2020).

4,6 Avaliação do desempenho da rede neural

Após o treinamento e os testes, é necessário analisar o desempenho da rede. A avaliação das respostas do aprendizado do modelo de rede neural proposto foi realizada por meio de uma matriz de confusão. A matriz de confusão possibilita relacionar a predição com a resposta real, de forma que as linhas indicam o que os padrões previstos, enquanto as colunas indicam as respostas reais (Artola, 2019). A Tabela 3 mostra uma matriz de confusão.

Tabela 3 - Matriz de confusão e seus elementos

	Verdadeiro	Falso
Verdadeiro	VP	FP
Falso	FN	VN

Os elementos da matriz representam as seguintes relações entre predição e realidade: Um verdadeiro positivo (VP) acontece quando a previsão é verdadeira e o real também; um falso positivo (FP) acontece quando a previsão é verdadeira e na realidade é falso; um falso negativo

(FN) acontece quando a previsão é falsa e na realidade é verdadeiro; um verdadeiro negativo (VN) acontece quando a previsão é falsa e a realidade também. Dessa forma, torna-se possível construir as seguintes métricas para análise do aprendizado da rede de acordo com Tabela 4.

Tabela 4 - Métricas para avaliação do aprendizado da rede proposta

A acurácia ou ***accuracy*** indica a proporção do número de predições corretas em relação ao número total de predições	$Accuracy = \frac{VP + VN}{VP + FN + FP + VN}$
A métrica precisão ou ***precision*** tem como objetivo identificar quantas amostras foram classificadas positivamente.	$Precision = \frac{VP}{VP + FP}$
A métrica revocação ou ***recall*** tem a mesma ideia do *precision*, porém para as amostras falsas negativas.	$recall = \frac{VP}{VP + FN}$
A métrica ***F1 score*** é definida como duas vezes a média harmônica entre a precisão (*precision*) e a revocação *(recall*)	$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$

Por outro lado, o "*loss*" (perda), é uma soma dos erros cometidos para cada exemplo em conjuntos de treinamento ou validação. Mostra o quão longe se está da solução 'ideal'. Quanto menor a **perda,** melhor o modelo, está métrica e sua interpretação é do desempenho no treinamento e na validação. Diferentemente da precisão, a perda não é uma porcentagem.

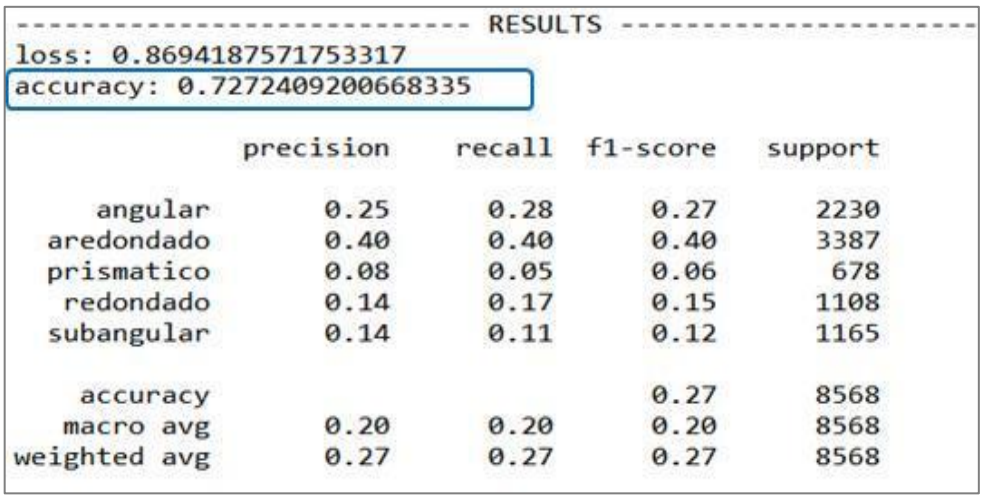
5. RESULTADOS E DISCUSSÕES

Com a finalidade de avaliar com maior detalhe o desempenho do modelo MobilenNetV2 se utilizaram 3 tipos de experimentos, Experimento 1 com 5 classes, Experimento 2 com 4 classes e Experimento 3 com 3 classes morfométricas de agregados do solo, cada uma avaliada com as métricas de desempenho de uma rede neural. O número de épocas para os três experimentos foi de 20. Se fizeram treinos com 10, 15, 20 e 50 épocas, mas os resultados foram melhores com 20 épocas e 32 de batch_size. Se utilizarmos poucas épocas, podem surgir problemas de underfitting, ou seja, a rede não consegue expressar seu máximo aprendizado. Entretanto, se utilizarmos um conjunto maior de épocas, pode ocorrer o problema oposto, overfitting, ou seja, a rede busca padrões excedentes de maneira até ajustar "ruído" nos dados de treinamento, e não o sinal pleno, conforme (DEEP LEARNING BOOK, 2019).

5.1 Experimento 1

Os resultados das métricas de aprendizado da rede neural da classificação de forma de agregados no Experimento 1 com 5 classes são apresentados na Figura 22. A *accuracy* pode ser observada na Figura 25, com um valor de 73% e uma perda de 0.87.

Figura 25 - Métricas do desempenho da rede neural com 5 classes



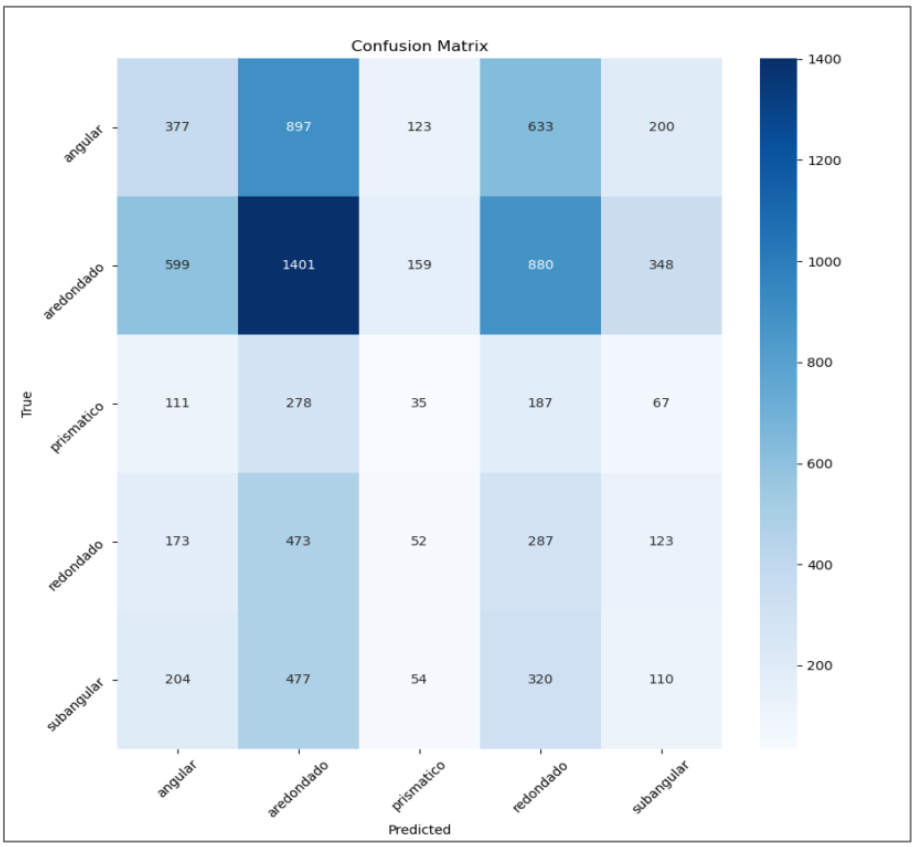
```
---------------------------- RESULTS -----------------------
loss: 0.8694187571753317
accuracy: 0.7272409200668335

              precision    recall  f1-score   support

     angular       0.25      0.28      0.27      2230
  aredondado       0.40      0.40      0.40      3387
  prismatico       0.08      0.05      0.06       678
   redondado       0.14      0.17      0.15      1108
  subangular       0.14      0.11      0.12      1165

    accuracy                           0.27      8568
   macro avg       0.20      0.20      0.20      8568
weighted avg       0.27      0.27      0.27      8568
```

Fonte: Autoria própria (2020).

Pela própria definição de *accuracy* como a proporção de casos que foram corretamente previstos, sejam eles verdadeiro positivo ou verdadeiro negativo, observa que a rede MobileNetV2 com 5 classes não apresentou um desempenho satisfatório. Essa condição de baixo desempenho se apresenta mais notória ainda, quando a base de dados para o treinamento da rede não é balanceada, por isso a importância do método *Under-Sampling* (Yen & Lee, 2016), não balancear as quantidades de imagens por classe pode apresentar uma performance boa para só um tipo de classe e não para a outra. Se observa na Figura 25 as outras métricas de desempenho principalmente a precisão por classes maior valor foi 40% para arredondado e 25% para angular. Com a interpretação visual da intensidade de cores da matriz de confusão, na Figura 26 observe-se que a intensidade maior ficou para a contagem arredondado. O ideal seria que a contagem se concentrasse na diagonal principal da matriz.

Dentro de uma análise comparativa do desempenho de aprendizado nas métricas da matriz de confusão com 5 classes (Figura 26), observa-se um melhor desempenho (precisão) da rede na classificação das classes Arredondada e Angular, que foram as classes com maior número de imagens, o que poderia haver influenciado na aprendizagem da rede. Entretanto, essa quantificação métrica se apresenta baixa, pois a precisão tem por finalidade identificar quantas amostras foram classificadas positivamente.

Figura 26 - Matriz de confusão com 5 classes



Fonte: Autoria própria (2020).

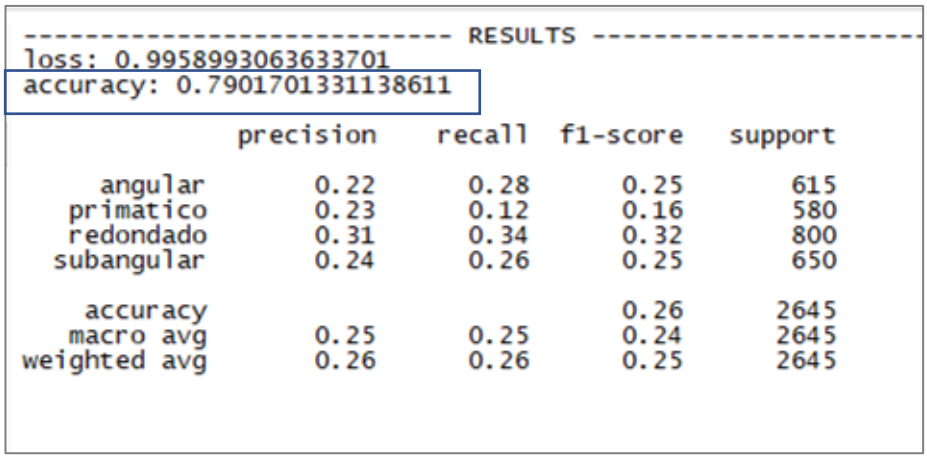
No primeiro experimento com 5 classes o desempenho de aprendizado da CNN foi baixo, isto devido ao desiquilíbrio das quantidades das imagens para cada classe, a classe que apresentou menor precisão foi a classe prismática, isto pode estar relacionada com ser a classe com menor número de imagens para o treinamento e validação com relação às outras classes. Assim, para melhorar o desempenho de aprendizado é necessária uma preparação de *data_set* mais homogênea e balanceada (QUINTERO et al., 2018). Nessa inserção, outro aspecto relevante, na melhoria do aprendizado, consiste na segmentação das imagens (SMITH, 2018). Entretanto, pode haver a necessidade de um maior número de amostras, pois a utilização de classes morfométricas de agregados semelhantes a metodologia de análise visual exige mais informações para uma melhor classificação e aprendizado (RODRIGUEZ, 2018).

5.2 Experimento 2

Observando-se o problema das heterogeneidades das imagens e o desbalanço do número de imagens por classes no Experimento 1 do aprendizado da rede, foram realizados os ajustes

da *data_set*, utilizando o método de *Under-Sampling*, assim para o Experimento 2 com 3 classes balanceadas, com 20 épocas e 32 de *batch_size*, se percebe-se uma melhora significativa no desempenho da classificação da CNN-MobileNetV2. O resultado do treinamento com o grupo de dados para teste mostrou um desempenho de *accuracy* de 79 % mas com uma perda de 0,99 maior do que o primeiro experimento. Entretanto, a quantificação métrica de precisão para cada classe também resultou com porcentagens mais similares, sendo maior para a classe redondeado y menor para a classe angular (Figura 27).

Figura 27 - Métricas do desempenho da rede neural com 4 classes

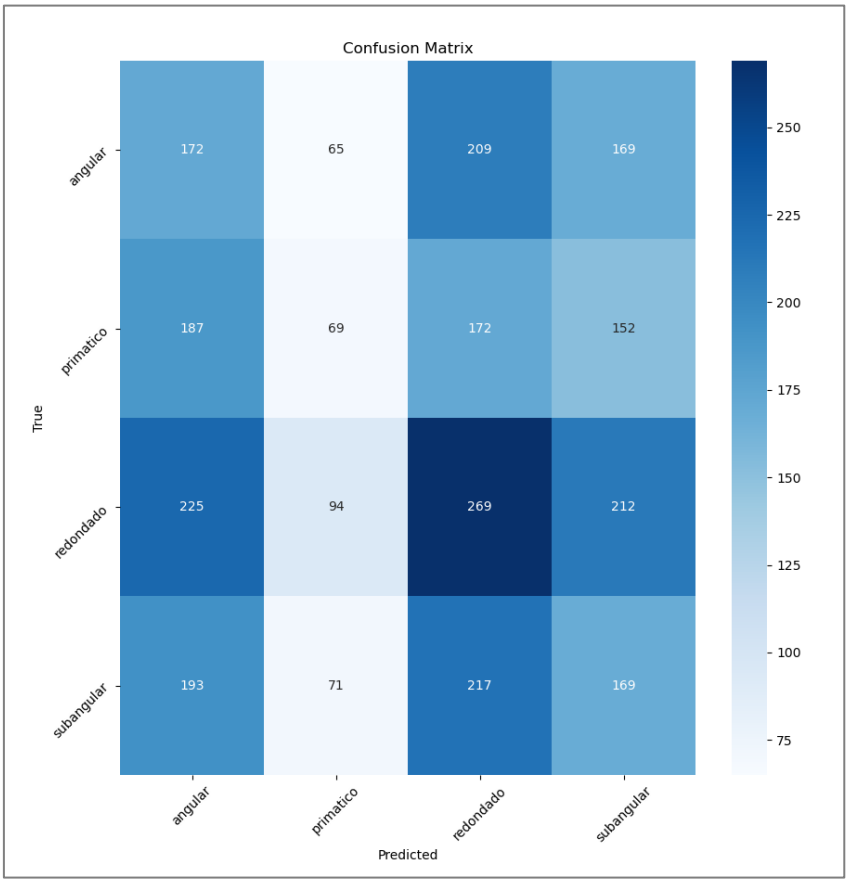


```
--------------------------- RESULTS ----------------------
loss: 0.9958993063633701
accuracy: 0.7901701331138611
              precision    recall  f1-score   support

     angular       0.22      0.28      0.25       615
   primatico       0.23      0.12      0.16       580
   redondado       0.31      0.34      0.32       800
  subangular       0.24      0.26      0.25       650

    accuracy                           0.26      2645
   macro avg       0.25      0.25      0.24      2645
weighted avg       0.26      0.26      0.25      2645
```

Fonte: Autoria própria (2020).

Dentro de uma análise da matriz de confusão do Experimento 2 com 4 classes, observa-se um melhor desempenho da rede na classificação das classes Redondo, Subangular e Angular. Entretanto, a precisão para a classe Angular e Prismática ainda se apresenta baixa, pois na Figura 28 observa-se os falsos positivos e falsos negativos com às outras três classes.

Figura 28 - Matriz de Confusão com 4 classes



Fonte: Autoria própria (2020).

Importante indicar que o método de análise visual requer conhecimentos especializados, conforme a descrição metodológica. Nesse contexto, a melhoria do aprendizado da rede neural também necessita da inserção de informações estáveis e métodos para maximizar a qualidade da informação de entrada e minimizar os erros no processo de aprendizado. A análise crítica do método visual auxilia o entendimento dos agregados do solo e fornece elementos notáveis para o uso de redes neurais (VAN LEEUWEN et al., 2018; BALL et al., 2017).

5.3 Experimento 3

Através da análise da Figura 29 é possível observar que o desempenho da rede do Experimento 3 com 3 classes, obteve um melhor desempenho comparado com os Experimentos 1 e Experimento 2, com uma *accuracy* de 87%. Para o Experimento 3 foi possível fazer um gráfico do desempenho da rede, relacionando a perda *(loss)* e a *accuracy* com as épocas de treinamento (Figura 30). A maior diferença foi na taxa de perda sendo 0.36, valor menor com a perda dos experimentos anteriores, esta melhora pode ser devido a melhora da qualidade da

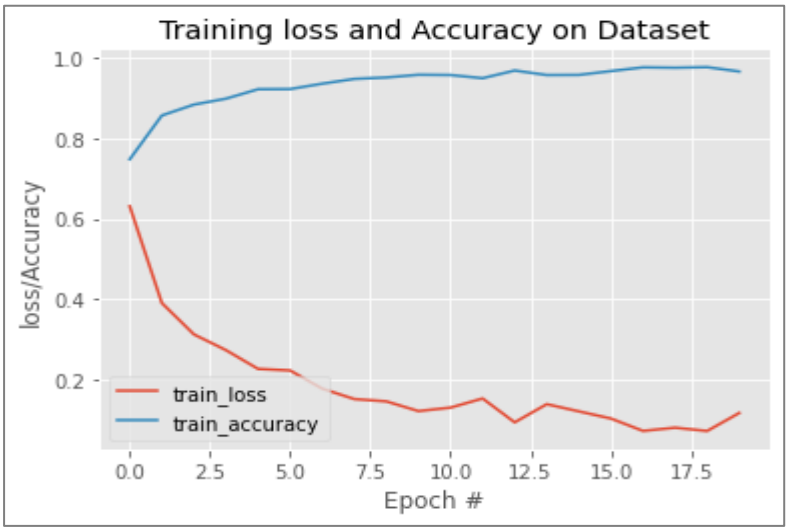
data_set, mas a diminuição do número de imagens por classe pode ter influenciado o processo de aprendizagem da rede (Anexo 5).

Figura 29 - Métricas do desempenho da rede neural com 3 classes

```
---------------------------- RESULTS ----------------------
loss: 0.36782321377712135
accuracy: 0.8710073828697205

              precision    recall  f1-score   support

     angular       0.37      0.35      0.36       557
  prismatico       0.36      0.34      0.35       555
   redondado       0.31      0.35      0.33       516

    accuracy                           0.35      1628
   macro avg       0.35      0.35      0.35      1628
weighted avg       0.35      0.35      0.35      1628
```

Fonte: Autoria própria (2020).

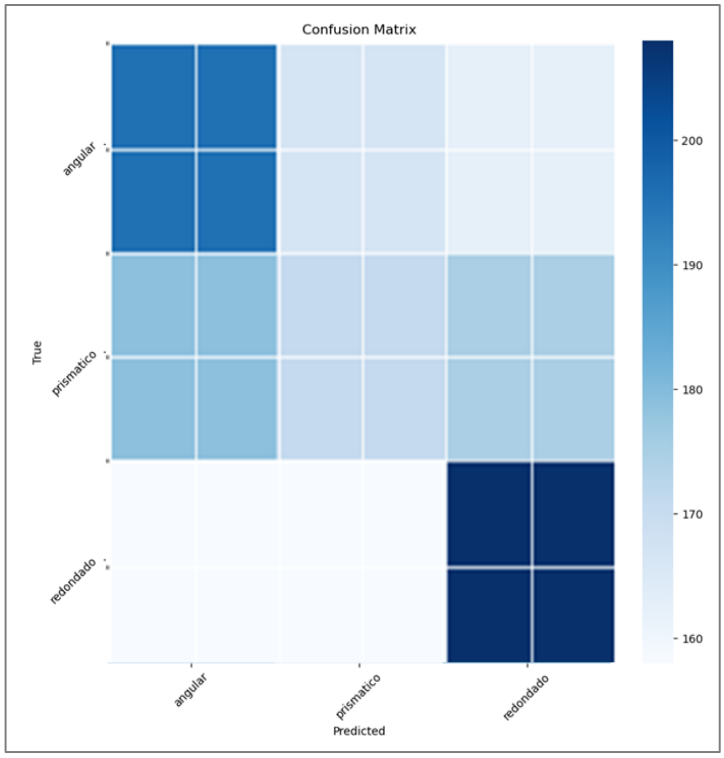
Figura 30 – Gráfico do treinamento da rede neural com 3 classes



Fonte: Autoria própria (2020).

Na análise da matriz de confusão do Experimento 3 com 3 classes, observa-se uma melhor precisão na classificação das classes Redondo e Angular. Entretanto, para a classe Prismática ainda se apresenta baixa, pois na Figura 31 observa-se os falsos positivos e falsos negativos com às outras duas classes.

Figure 31-Matriz de Confusão com 3 classes



Fonte: Autoria própria (2020).

Nos três experimentos com o modelo MobileNetV2 a *accuracy* media foi 79,3%, e a maior *accuracy* alcançada foi com o Experimento 3 com 3 classes morfométricas (87%), valores similares a *accuracy* de *87%* alcançada por um modelo multicamada de ANN desenvolvida para estimar o índice de estabilidade de agregados do solo com dados numéricos por TORKASHVAND & AUTHOR, (2017) e uma *accuracy* de 73,81% de um classificador baseado em redes neurais artificiais para gerar um mapa com diferentes classes de solo através da análise de imagens digitais satelitais desenvolvida por Chagas *et al*. (2013). Mas os resultados obtidos são menores comparados com a *accuracy* alcançada dos modelos de CNN VggNet16, ResNet50 e Inception-v4, trabalhada por AZZI *et al.* (2020) utilizando imagens estéreo para classificar os agregados por tamanho, com uma média de *accuracy* de 95% sendo a maior *accuracy* alcançada com a arquitetura ResNet50 (98,72%).

A pesquisa realizada por AZZI *et al.* (2020) tem certa semelhança com a pesquisa desenvolvida, mas eles utilizaram várias CNNs também realizando transferência de aprendizagem, em média 100 épocas utilizando uma GPU, classificando os agregados por tamanho em 6 classes, em nosso caso, foi para classificar os agregados por morfometria (5 classes, 4 classes e 3 classes). Uma maior diferencia é no tipo de aquisição das imagens, eles

utilizaram uma câmera estereofônica equipada com dois sensores com captura de imagem 3D, isto pode influenciar na qualidade das imagens principalmente nas mudanças da luz do ambiente.

Em quanto a método utilizado para o estudo dos agregados do solo por meio de imagens digitais, tem suas vantagens e desvantagens, uma vantagem dos métodos de avaliação do solo mediante imagem, é que agregado por exemplo não sofre altercações físicas, químicas nem biológicas comparado com os analises nos laboratórios tradicionais (Padarian, Minasnya & Mcbratneya, 2018), más sofrem de limitações como Software, hardware, quantidade e qualidade de obtenção das imagens, o grau de subjetividade nos protocolos de processamento de imagem, a extração de características e o requerimento de algum nível de informação do utilizador em termos de dados de formação e aprendizagem profundo (SCHLÜTER *et al.*, 2020), influenciando assim no desempenho nos resultados da rede neuronal.

As CNN possuem múltiplos usos e muitas potencialidades de aplicação na classificação de imagens nas diferentes áreas do conhecimento humano, mas é importante observar que técnicas eficazes de aprendizado de máquina geralmente requerem muitos dados para treinamento. Do Experimento 1 (30450 imagens) até o Experimento 3 (9220 imagens) foram diminuindo a quantidade das imagens para o treinamento e validação, devido aos processamentos e métodos para melhorar a qualidade da *data_set*. Portanto, a coleta de dados deve ser maior, uma quantidade que permita ser significante despois de fazer o pré-processamento e ter aplicado os métodos para balancear e homogeneizar os dados por classe.

Através desta pesquisa e de outras pesquisas, como classificar diferentes grupos de solos degradados (RIBEIRO *et al.*,2016), identificação de áreas erodidas (TRUEVA *et al*., 2004), predição de classes de solos (CHAGAS *et al*., 2013), prever as propriedades do solo (Padarian, Minasnya & Mcbratneya, 2018), classificação de tamanhos de agregados dos solos (AZIZI et al., 2020) e análise da cor do solo (AL-NAJI *et al*., 2021), as redes neurais artificiais, são consideradas como uma ferramenta, uma alternativa nas ciências ambientais e agrícolas, que pode ajudar a otimizar o estudo dos solos em geral, para sua gestão e recuperação, para uma agricultura de precisão.

Independentemente do baixo desempenho de aprendizado pela rede neural exemplificada, seja do Experimento 1, Experimento 2, Experimento 3. O caminho se apresenta promissor, pois assim como na análise visual o treinamento de uma rede neural pode ser capaz de analisar os parâmetros de rugosidade dos agregados, forma, atividade biogênica e outros, sendo de grande contribuição para a gestão estrutural do solo de forma remota e inteligente.

6. CONCLUSÕES

A avaliação do aprendizado rede neural do modelo MobileNetV2 para os três experimentos mostraram uma *accuracy* média de 79%, sendo mais satisfatória para a classificação morfométrica do agregado do solo das classes redondado e angular, ou seja, uma condição de extremos.

A utilização dos métodos de *data_augmentation, Under sampling* e *data_cleaning* influenciam nos resultados do desempenho, métodos utilizados para o Experimento 2 e Experimento 3, onde a rede MobileNetV2 chegou a alcançar uma *accuracy* de 87%.

A preparação da *data_set* consiste em uma fase muito importante, como foi analisado nos três experimentos da dissertação. Essa preparação pode fazer a diferença no desempenho do aprendizado da rede, pois está associado aos seguintes fatores: qualidade e quantidade de imagens por classe, estratégia de pré-processamento em geral, escolha do hardware e software e o nível de conhecimento do objeto em estudo.

7. REFERÊNCIAS

Aguilar, J., Dorronsoro-Fdez, C., Fernández, J., Dorronsoro Diaz, C., Martin, F., & Dorronsoro, B. (2019). *Micromorfografía de suelos. Indice*. Http://edafologia.ugr.es/micgraf/index.htm

Anghinoni, I., Carvalho, P. D. F., & Costa, S. D. A. (2013). *Abordagem sistêmica do solo em sistemas integrados de produção agrícola e pecuária no subtrópico brasileiro.* Tópicos em Ciência do Solo, 8(2), 325-380.

Artola, Á. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python* [Escuela Técnica Superior de Ingeniería Universidad de Sevilla]. Http://bibing.us.es/proyectos/abreproy/92402/fichero/TFG-2402-ARTOLA.pdf

Baker, B., Gupta, O., Naik, N., & Raskar, R. (2017, November 7). Designing neural network architectures using reinforcement learning. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. Https://bowenbaker.github.io/metaqnn/

Bini, D., Varon-Lopez, M., & Nogueira, E. (2016, January). *Metabolismo Microbiano*. Https://www.researchgate.net/publication/311788702_Metabolismo_Microbiano

Brady, N. C., & Weil, R. R. (2009). Elementos da natureza e propriedades dos solos. Bookman Editora.

Castro Filho, C., Muzilli, O., & Podanoschi, A. L. (1998). *Estabilidade dos agregados e sua relação com o teor de carbono orgânico...* 527 Estabilidade dos agregados e sua relação com o teor de carbono orgânico num latossolo roxo distrófico, em função de sistemas de plantio, rotações de culturas e métodos de preparo das amostras *(1)* (Vol. 22).

Catafesta Francisco, Rodrigo. (2014). Estudo Da Viabilidade Técnica De Produção Do Concreto De Alto Desempenho Utilizando Agregados Encontrados Na Região Do Vale Do Itajaí. 10.13140/RG.2.1.4119.8563.

Salton, Júlio Cesar, Mielniczuk, João, Bayer, Cimélio, Boeni, Madalena, Conceição, Paulo Cesar, Fabrício, Amoacy Carvalho, Macedo, Manuel Cláudio Motta, & Broch, Dirceu

Luiz. (2008). Agregação e estabilidade de agregados do solo em sistemas agropecuários em Mato Grosso do Sul. Revista Brasileira de Ciência do Solo, 32(1), 11-21. Https://doi.org/10.1590/S0100-06832008000100002

Chagas, C. Da S., Vieira, CAO, & Fernandes Filho, EI (2013). Comparação entre redes neurais artificiais e classificação por máxima verossimilhança sem mapeamento digital de solos. Revista Brasileira de Ciência do Solo , 37 (2), 339–351. Https://doi.org/10.1590/S0100-06832013000200005

Chitero, Jgm; Bonini Neto, A .; Bonini, C. Dos Sb; Heinrichs, R .; Soares Filho, Cv; Mateus, Gp; Bisi, Bs; Costa, Nr; Piazentin, Jc; Meirelles, Gc; Gabriel Filho, Lra Análise da recuperação física de solos degradados via Redes Neurais Artificiais utilizando uma interface gráfica. Pesquisa, Sociedade e Desenvolvimento , [S. L.] , v. 9, n. 7, pág. E257973719, 2020. DOI: 10.33448 / rsd-v9i7.3719. Disponível em: https://rsdjournal.org/index.php/rsd/article/view/3719. Acesso em: 26 jan. 2021.

De Carvalho, F., Diniz, F., Savana, J., Valentim, J., de Paiva, L., Soares, T., de Melo, W., Palhares, T., & de Souza, F. (2017). *Micro organismos* (M. Toma, R. Vilas, & F. De Souza, Eds.; UFLA). Www.editora.ufla.br

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. In *Progress in Artificial Intelligence* (Vol. 9, Issue 2, pp. 85–112). Springer. Https://doi.org/10.1007/s13748-019-00203-0

Durafour, M., Jarno, A., le Bot, S., Lafite, R., Marin, F., Bedload, M., & Lafite, · Robert. (2015). *Bedload transport for heterogeneous sediments*. *15*(4), 731–751. Https://doi.org/10.1007/s10652-014-9380-1ï

Hadican. (2019, March 28). *How to Build a Simple Artificial Neural Network (ANN)*. Https://medium.com/@hadican/how-to-build-a-simple-artificial-neural-network-ann-a064939f940b

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *arxiv*. Arxiv. Https://arxiv.org/abs/1704.04861v1

Izaurieta, F., & Saavedra, C. (2000). Redes neuronales artificiales. Departamento de Física, Universidad de Concepción Chile.

Jan, B., Farman, H., Khan, M., Imran, M., Islam, I. U., Ahmad, A., Ali, S., & Jeon, G. (2019). Deep learning in big data Analytics: A comparative study. *Computers and Electrical Engineering*, *75*, 275–287. Https://doi.org/10.1016/j.compeleceng.2017.12.009)

FAO (2016). Estado Mundial del Recurso Suelo (EMRS) – Resumen Técnico, Roma, Italia.

Gervasio Pereira, M., Helena Cunha dos Anjos, L., Roberto Pinheiro Junior, C., Alberto da Silva Rodrigues Pinto, L., Carvalho da Silva Neto, E., & Fontana, A. (2019). *CAPÍTULO 1 FORMAÇÃO E CARACTERIZAÇÃO DE SOLOS*.

Grubler, M. (2018, May 11). *Entendendo o funcionamento de uma Rede Neural Artificial | by Murillo Grübler | aibrasil | Medium*. Https://medium.com/brasil-ai/entendendo-o-funcionamento-de-uma-rede-neural-artificial-4463fcf44dd0

Guo, T., Dong, J., Li, H., & Gao, Y. (2017). Simple convolutional neural network on image classification. *2017 IEEE 2nd International Conference on Big Data Analysis, ICBDA 2017*, 721–724. Https://doi.org/10.1109/ICBDA.2017.8078730

Guerra, A. T., da Silva, A. S., & Botelho, R. G. M. (2009). Erosão e conservação dos solos: conceitos, temas e aplicações. Bertrand Brasil.

Kohn, L. S. (2017). Análise morfométrica de agregados de um Cambissolo Húmico sob o cultivo de linho e seu desenvolvimento radicular.

León, M. P., & Ramírez, F. (2010). Morphological characterization of concrete aggregates by means of image analysis. *Revista Ingenieria de Construccion*, *25*(2), 215–240. https://doi.org/10.4067/S0718-50732010000200003

López, R. F., & Fernández, J. M. F. (2008). Las redes neuronales artificiales. Netbiblo.

MATLAB (2020) 9.7.0.1190202 (r2019b). Natick, Massachusetts: The mathworks Inc.

Nogueira, E., & Dini, F. (2016). 済無No Title No Title. In *Journal of Chemical Information and Modeling* (3rd ed., Vol. 53, Issue 9Matich, D. J. (2001). Redes Neuronales: Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional, México.

Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Guerrero Méndez, C. A., & Olvera-Olvera, C. A. (2020). Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases. *Applied Sciences*, *10*(4), 1245. Https://doi.org/10.3390/app10041245

MOREIRA, F.M.S.; SIQUEIRA, J.O., 2006. Microbiologia e Bioquímica do solo. Editoraufla. Lavras – MG, 2ªedição, 729p.

PECHE FILHO, A. (2018). Variabilidade da agregação em amostras de solos agrícolas como indicador de qualidade: uma proposta metodológica. Tesis doctoral. Universidade Estadual Paulista "Júlio de Mesquita Filho"

Olabe, X. B. (1998). Redes neuronales artificiales y sus aplicaciones. Publicaciones de la Escuela de Ingenieros.

Pérez-Pérez, Alejandro, Matadamas-Ortiz, Idarh Claudio, Morales-Hernández, Maricela Y Altamirano-Cabrera, Marisol. Prototipo basado en Redes Neuronales para monitoreo y predicción de temperatura en invernaderos de los Valles Centrales de Oaxaca.Revista de Prototipos Tecnológicos 2017.

Ponti, M. A., & da Costa, G. B. P. (2018). *Como funciona o Deep Learning*. Http://arxiv.org/abs/1806.07908

PUJARA, A. (2020, July 4). *Rede neural convolucional mobilenet Algoritmos de aprendizado de máquina | Analytics Vidhya*. Https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470

Quintero, C., Merchán, F., Cornejo, A., & Galán, J. S. (2018). Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macroinvertebrados para el Biomonitoreo Participativo. Kne Engineering, 585-596.

Rabot, E., Wiesmeier, M., Schlüter, S., & Vogel, H. J. (2018). Soil structure as an indicator of soil functions: A review. In *Geoderma* (Vol. 314, pp. 122–137). Elsevier B.V. https://doi.org/10.1016/j.geoderma.2017.11.009

Ralisch, R., Debiasi, H., Franchini, J. C., Tomazi, M., Hernani, L. C., Melo, A. Da S. ; S. A., Martins, A. L. Da S., & De Bona, F. D. (2017). *Diagnóstico rápido da estrutura do solo - DRES* (EMBRAPA). Embrapa Soja. Https://www.embrapa.br/soja/busca-de-publicacoes/-/publicacao/1071114/diagnostico-rapido-da-estrutura-do-solo---dres-livro

Rodriguez, V. (2018, October 30). *Noções básicas de rede neural*. Https://vincentblog.xyz/posts/conceptos-basicos-sobre-redes-neuronales

Salton, J. C., Mielniczuk, J., Bayer, C., Boeni, M., Conceição, P. C., Fabrício, A. C., Macedo, M. C. M., & Broch, D. L. (2008). Soil aggregation and aggregate stability under crop-pasture systems in Mato Grosso do Sul state, Brazil. *Revista Brasileira de Ciencia Do Solo, 32*(1), 11–21. Https://doi.org/10.1590/s0100-06832008000100002

Serna, E. (2017). Desarrollo e Innovación en Ingeniería (2nd ed.). Editorial Instituto Antioqueño de Investigación. Https://d1wqtxts1xzle7.cloudfront.net/59788956/2017Desarrolloeinnovacineningenieria220190618-76386-og5bni.pdf?

Suarez Arnol, Jiménez Andrés, Castro Franco Mauricio y Cruz-Roa Ángel. Clasificación automática de coberturas del suelo en imágenes satelitales utilizando redes neuronales convolucionales: Un caso aplicado en Parques Nacionales Naturales de Colombia, 2016.

Souza, Lucas & Silva, Érika & Oliveira, Geraldo & Barbosa, Samara & Silva, Bruno. (2018). Análise Qualitativa E Quantitativa De Agregados De Solo Sob Mulching Plástico Associado À Aplicação De Fertilizantes Químicos E Organominerais Em Área Cafeeira. Scientia Agraria. 19. 142-153.

Sharma, N., Jain, V., & Mishra, A. (2018). An Analysis of Convolutional Neural Networks for Image Classification. Procedia Computer Science, 132, 377–384. Https://doi.org/10.1016/j.procs.2018.05.198

Silva, Marx & Freitas, Diego & Cândido, Bernardo & Oliveira, A.. (2015). Manejo e conservação do solo e da água - guia de estudos. 10.13140/RG.2.1.3609.2241.

Sudeep, K. S., & Pal, K. K. (2017). Preprocessing for image classification by convolutional neural networks. 2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 - Proceedings, 1778–1781. Https://doi.org/10.1109/RTEICT.2016.7808140

Smith, L. N. (2018). A Disciplined Approach To Neural Network Hyper-Parameters: Part 1 – Learning Rate, Batch Size, Momentum, And Weight Decay. In arxiv. Arxiv. Https://github.com/lnsmith54/hyperparam1.

Steinbeck, J. V. (2006). Arquitetura e Propiedades Físicas do Solo. Https://edisciplinas.usp.br/pluginfile.php/917327/mod_resource/content/2/Apostila%20-%20Arquitetura%20e%20Propriedades%20F%C3%adscas%20do%20Solo.pdf

Tensorflow. (2020). Https://www.tensorflow.org/

Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). Deep convolution neural network for image recognition. Ecological Informatics, 48, 257–268. Https://doi.org/10.1016/j.ecoinf.2018.10.002

Vila, E. M. S., & Penín, M. L. (2007). Monografía: Técnicas de la Inteligencia Artificial aplicadas a la educación. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 11(33), 7-12.

Yen SJ., Lee YS. (2006) Under-Sampling Approaches for Improving Prediction of the Minority Class in a Imbalanced Dataset. Em: Huang DS., Li K., Irwin GW (eds) Intelligent Control and Automation. Lecture Notes in Control and Information Sciences, vol. 344. Springer, Berlin, Heidelberg. Https://doi.org/10.1007/978-3-540-37256-1_89

8. ANEXOS

Anexo 1- Algoritmo do treinamento da CNN

```
"""Script to load a pretrained model and update all layers

In this script, the MobileNetV2 is loaded with imagenet weights, and
retrained with train and test samples
"""
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
import scipy
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

tf.autograph.set_verbosity(3, True)
# load the MobileNetV2 model with imagenet weights, not including the top part of the model
model = tf.keras.applications.MobileNetV2(weights='imagenet', include_top=False)
#model = tf.keras.applications.MobileNetV2(include_top=False)


# appends top layers to the model
x = model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dropout(0.25)(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dropout(0.25)(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dropout(0.25)(x)
x = tf.keras.layers.Dense(512, activation='relu')(x)
x = tf.keras.layers.Dropout(0.25)(x)
preds = tf.keras.layers.Dense(3, activation='softmax')(x)

# creates a new model with base of MobileNetV2 and the custom top part
model = tf.keras.models.Model(inputs=model.input, outputs=preds)
# prints the summary of the models (i.e, all layers and sizes)
model.summary()

# set all layers to be trainable
for l in model.layers:
        l.trainable = True

# creates train and test generators
train_data_gen = tf.keras.preprocessing.image.ImageDataGenerator(
   preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input)
test_data_gen = tf.keras.preprocessing.image.ImageDataGenerator(
   preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input)

train_generator = train_data_gen.flow_from_directory('dataset\\traintres',
   target_size=(224, 224), color_mode='rgb', batch_size=32,
   class_mode='categorical', shuffle=True)
test_generator = test_data_gen.flow_from_directory('dataset\\testtres',
   target_size=(224, 224), color_mode='rgb', batch_size=32,
   class_mode='categorical', shuffle=True)

train_steps = train_generator.n//train_generator.batch_size
test_steps = test_generator.n//test_generator.batch_size
```

```
# compile the model
print("compile the model")
model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['accuracy'])

# starts training for 20 epochs
print("start training")
model.fit_generator(generator=train_generator, steps_per_epoch=train_steps, epochs=20,
validation_data=test_generator, validation_steps=test_steps)

# saves the model in json format
print("save the model")
model_json = model.to_json()
with open('mobilenetv2_model2.json', 'w+') as f:
  f.write(model_json)

# saves the weights in h5 format
print("save waight")
model.save_weights('mobilenetv2_weights2.h5')
model.save('mobilenetv2')

print("Training")
Y_pred = model.predict_generator(test_generator, test_generator.n//test_generator.batch_size+1)
y_pred = np.argmax(Y_pred, axis=1)
cfMatr = confusion_matrix(test_generator.classes, y_pred)
print(cfMatr)
#target_names = ['angular', 'aredondado', 'prismatico', 'redondado', 'subangular'] #classes names
#target_names = ['angular', 'prismatico', 'redondado', 'subangular']
target_names = ['angular', 'prismatico', 'redondado']

clasRpt = classification_report(test_generator.classes, y_pred, target_names=target_names)
print(clasRpt)
```

ANEXO 2- Algoritmo do teste da CNN

```
from sklearn.metrics import confusion_matrix, classification_report
from datetime import datetime

import matplotlib.pyplot as plt
import tensorflow as tf
import seaborn as sb
import pandas as pd
import numpy as np
import pickle
import sys
import os

model_file = sys.argv[1]
model_weights_file = sys.argv[2]

if not os.path.isdir('outputs'):
    os.mkdir('outputs')

now = datetime.now()
now_str = f'{now.year}_{now.month}_{now.day}T{now.hour}_{now.minute}_{now.second}'
if not os.path.isdir(os.path.join('outputs', now_str)):
    os.mkdir(os.path.join('outputs', now_str))


with open(model_file, 'r') as f:
    json_model = f.read()
model = tf.keras.models.model_from_json(json_model)

model.load_weights(model_weights_file)
model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

test_data_gen = tf.keras.preprocessing.image.ImageDataGenerator(
    preprocessing_function=tf.keras.applications.mobilenet.preprocess_input)

test_generator = test_data_gen.flow_from_directory('dataset\\test_tres',
    target_size=(224, 224), color_mode='rgb', batch_size=32
    ,
    class_mode='categorical', shuffle=True)


# stores metrics, predictions and true labels
metrics = model.metrics_names
scores = model.evaluate_generator(test_generator)
predictions = model.predict_generator(test_generator)
predicted_classes = np.argmax(predictions, axis=1)
true_classes = test_generator.classes
classes = test_generator.class_indices
cm = confusion_matrix(true_classes, predicted_classes)
test_data = {
    "metrics": metrics,
    "scores": scores,
    "predictions": predictions,
    "predicted_classes": predicted_classes,
    "classes": classes,
    "true_classes": true_classes,
    "cm": cm
}
```

```
data_file = open(f'outputs/{now_str}/test_data.p', 'wb')
pickle.dump(test_data, data_file)
data_file.close()

classes = list(classes.keys())
cm_df = pd.DataFrame(cm, index=classes, columns=classes)

fig = plt.figure(figsize=(10,10))
sb.heatmap(cm_df, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
ax, _ = fig.get_axes()
ax.set_ylabel('True')
ax.set_xlabel('Predicted')
ax.set_yticklabels(ax.get_yticklabels(), rotation=45)
ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
plt.savefig(f'outputs/{now_str}/cm.png')

report = '---------------------------- RESULTS ----------------------------\n'
report += '\n'.join([f'{metrics[i]}: {scores[i]}' for i in range(len(scores))]) + '\n\n'
report += classification_report(true_classes, predicted_classes, target_names=classes)
print(report)

with open(f'outputs/{now_str}/report.txt', 'w+') as f:
    f.write(report)
print()
```

Anexo 3 – Algoritmo do redimensionamento das imagens

```
Reads all images and resizes them to 480 x 480

Params
    root (str): root path to look for folders containing images

Example:
    $> python3 redims_images.py images
Where `images` is a folder that contain folder for each class with images
"""
import cv2
import sys
import os

# root path param
root = sys.argv[1]

# read all folders
folders = [item for item in os.listdir(root) if os.path.isdir(os.path.join(root, item))]

# for each folder, read all files in the folder
for folder in folders:
    files = [item for item in os.listdir(os.path.join(root, folder)) if os.path.isfile(os.path.join(root, folder, item))]
    total = len(files)
    print(folder)
    # for each image in the folder
    for i, image in enumerate(files):
        # prints the current image and total left
        print('    [{current:5d}/{total:5d}]'.format(current=i + 1, total=total), end='\r')
        # reads the image
        img = cv2.imread(os.path.join(root, folder, image))
        # if it could read the image, resizes it to 480 x 480
        if img is not None:
            resized = cv2.resize(img, (480, 480), interpolation=cv2.INTER_AREA)
            cv2.imwrite(os.path.join(root, folder, image), resized)
    # new line for the next folder

    print()
```

Anexo 4- Algoritmo da *data_augmentation.*

```
"""Expand a set of images by applying data augmentation

Creates new images by applying the following:
   rotations:
      90 degrees
      180 degrees
      270 degrees
   flip:
      original images
      90 degrees rotated images
      180 degrees rotated images
      270 degrees rotated images

Resulting in 7 new images for each image in the set.


This script should be applied in the train set ONLY!

Params:
   root (str): root path to search for folders containing images

Example:
   $> python3 data_augmentation.py dataset/train
Where `dataset/train` is a folder that contain train set's folders for
each class with images
"""
import cv2
import sys
import os

# root path param
root = sys.argv[1]

# read all folders
folders = [item for item in os.listdir(root) if os.path.isdir(os.path.join(root, item))]
# load the angles to rotate the images
angles = [90, 180, 270]

# for each folder, read all images
for folder in folders:
   print(folder + ':')
   files = [item for item in os.listdir(os.path.join(root, folder)) if os.path.isfile(os.path.join(root, folder, item))]
   total = len(files)
   # for each image
   for i, image in enumerate(files):
      # prints the current image and the total images in the folder
      print('   [{current:6d}/{total:6d}]'.format(current=i, total=total), end='\r')

      # reads the image and calculates the shape of it
      img = cv2.imread(os.path.join(root, folder, image))
      height, width = img.shape[:2]

      # calculates the horizontal and vertical center
      center_horizontal = (width / 2, height / 2)
      center_vertical = (height / 2, width / 2)

      # flips the original image and saves it
      flip_0 = cv2.flip(img, 1)
```

```
        basename = os.path.splitext(image)[0]
        cv2.imwrite(os.path.join(root, folder, basename + '_flipped_0.jpg'),
            flip_0)

        # rotates the original image to 90 degrees, saves it, flips the rotated and saves it
        original_rotated = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
        cv2.imwrite(os.path.join(root, folder, basename + '_original_90.jpg'),
            original_rotated)

        flipped_rotated = cv2.rotate(flip_0, cv2.ROTATE_90_COUNTERCLOCKWISE)
        cv2.imwrite(os.path.join(root, folder, basename + '_flipped_90.jpg'),
            flipped_rotated)

        # rotates the original image to 180 degrees, saves it, flips the rotated and saves it
        original_rotated = cv2.rotate(original_rotated, cv2.ROTATE_90_COUNTERCLOCKWISE)
        cv2.imwrite(os.path.join(root, folder, basename + '_original_180.jpg'),
            original_rotated)

        flipped_rotated = cv2.rotate(flipped_rotated, cv2.ROTATE_90_COUNTERCLOCKWISE)
        cv2.imwrite(os.path.join(root, folder, basename + '_flipped_180.jpg'),
            flipped_rotated)

        # rotates the original image to 270 degrees, saves it, flips the rotated and saves it
        original_rotated = cv2.rotate(original_rotated, cv2.ROTATE_90_COUNTERCLOCKWISE)
        cv2.imwrite(os.path.join(root, folder, basename + '_original_270.jpg'),
            original_rotated)

        flipped_rotated = cv2.rotate(flipped_rotated, cv2.ROTATE_90_COUNTERCLOCKWISE)
        cv2.imwrite(os.path.join(root, folder, basename + '_flipped_270.jpg'),
            flipped_rotated)
    # prints new line for the next folder

    print()
```

Anexo 5- print das imagens do teste com sua predição (azul os corretos e vermelho os incorretos)