

Visão Computacional aplicado na identificação do estado de telas dos filtros verticais no processo de secagem do minério

Leonardo Souza Coelho * Giovanni Bernardes Vitor *
Willian Gomes de Almeida * Rafael Francisco dos Santos *
Paulo Henrique Vieira Soares **

* *Laboratório de Robótica, sistemas inteligentes e Complexos - RobSIC
Universidade Federal De Itajubá - UNIFEI - Itabira*

** *Gerência de Serviço de Tecnologia de Automação Sudeste, VALE
S/A
Itabira, Minas Gerais, Brasil.*

Abstract: Industry 4.0 has provided significant advances in several areas, one of them being in the mining sector. Within the mining process, the ore drying stage is characterized as an important phase in which the ore passes through rotating vacuum filters, in order to reduce its moisture. One of the problems at this stage concerns the detection of the quality of filters in the drying machine, with the clogging of the filtering medium being the main reason for the poor quality of the filter. Therefore, this work proposes the use of deep neural networks to recognize the quality of these filters for preventive maintenance. Unet and SegNet network architectures were applied, two types of convolutional neural networks, *encoder/decoder*. As a result, it was possible to obtain more than 90% accuracy in the correctness of the model, as well as the determination of a quality indicator for these equipments.

Resumo: A indústria 4.0 têm proporcionado significativos avanços em diversas áreas, sendo uma delas o setor de mineração. Dentro do processo de mineração, a etapa de secagem do minério caracteriza-se como uma importante fase a qual o minério passa por filtros, como os filtros rotativos à vácuo, com o objetivo de diminuir sua umidade. Um dos problemas existentes nessa etapa diz respeito a detecção da qualidade dos filtros existentes na máquina de secagem, sendo o entupimento do meio filtrante o principal motivo da má qualidade do mesmo. Logo, este trabalho propõe o uso de redes neurais profundas para classificar e mensurar a qualidade desses filtros para realização de manutenção preventiva. Foram aplicadas as arquiteturas de redes Unet e SegNet, dois modelos de redes neurais convolucionais do tipo *encoder/decoder*. Como resultado, foi possível obter mais de 90% de precisão no acerto do modelo, bem como a determinação de um indicador de qualidade das telas verticais relacionadas a esses equipamentos.

Keywords: Ore Drying; Artificial Intelligence; Computer Vision; Automation 4.0; Mining;

Palavras-chaves: Secagem de Minério; Inteligência Artificial; Visão Computacional; Automação 4.0; Mineração;

1. INTRODUÇÃO

A indústria 4.0 chega como um grande avanço tecnológico no meio industrial, trazendo inúmeras inovações, impactando diversos setores, mostrando otimizações e agilidades em todos os processos. Em seu estudo sobre os impactos da indústria 4.0 na mineração, Cesário Neto (2020) ressalta que a indústria da mineração busca ter uma melhora na eficiência produtiva, redução em danos físicos e ambientais, e uma melhor capacidade de planejamento.

O processo de beneficiamento do minério de ferro exigem várias etapas que começam na perfuração e desmonte das rochas na mina até atingir o produto final que será vendido. O processo é dividido nas etapas: cominuição (britagem e moagem); peneiramento e classificação; concentra-

ção; desaguamento (espessamento e filtragem); secagem; e disposição de rejeito (Luz and Lins, 2018).

Durante a etapa final do desaguamento é realizado o processo de filtragem na polpa de minério de ferro obtida nas etapas anteriores. O processo de filtragem é responsável em separar o minério de ferro da água, onde partículas de minério são acumuladas no exterior do meio filtrante formando "tortas". Vários tipos de filtros podem ser aplicados, como filtros rotativos a vácuo (tambor e disco), filtros de correia, filtro-prensa e filtros de pressão (hiperbáricos)(Luz and Lins, 2018). Este trabalho terá como foco os filtros rotativos de disco vertical a vácuo.

Soares et al. (2017), traz uma pesquisa que busca analisar e observar os problemas durante o processo de filtragem usando filtros de disco vertical a vácuo. Erros nesta etapa,

causam a paralisação da filtragem, o que acaba impactando diretamente a produção.

Um dos problemas se mostra no entupimento do meio filtrante que prejudica a retirada da umidade da polpa. Esse problema torna o processo menos eficiente criando a má formação das tortas nos filtros. Exemplos de qualidade dessas tortas podem ser vistos na Figura 1, onde em (a) possui um filtro com má formação e em (b) com boa formação.



Figura 1. Qualidade na formação da torta. (a) possui um filtro com má formação e em (b) com boa formação.

Tendo em vista a necessidade de classificar e mensurar a qualidade do meio filtrante, este trabalho tem como proposta utilizar tecnologias de inteligência artificial, para obter uma forma de classificação que seja satisfatória e forneça informações para uma abordagem preditiva, atenuando os possíveis impactos na produção.

Na sequência do trabalho, encontra-se o estado da arte na Seção 2, mostrando estudos feitos na área e suas aplicações. A Seção 3 consiste no processo de desenvolvimento do trabalho, evidenciando cada etapa realizada. Em seguida, na Seção 4 encontram-se os resultados adquiridos, e por fim, na Seção 5, temos a conclusão e as sugestões para trabalhos futuros.

2. ESTADO DA ARTE

Visando o processo de filtragem do minério, os estudos a seguir expõem algumas possíveis soluções para o problema apresentado.

De acordo com Oliveira et al. (2004) um processo de filtragem na mineração consiste em separar o sólido-líquido através de um meio que retém os sólidos e permite a passagem do líquido. Em uma filtragem típica existe uma camada denominada torta, onde há o acúmulo de sólido aglomerado, a polpa a ser filtrada e o líquido separado. Para aumentar a eficiência é preciso que o meio permita a passagem do líquido com a mínima resistência, sem que seja permitida a passagem dos sólidos. Os materiais que podem ser utilizados na fabricação do meio filtrante são, algodão, lã, nylon, seda, juta, fibra de vidro, borracha porosa, carbono poroso, metais e outros materiais sintéticos. O estudo apresenta que um fator significativo para o entupimento é a quantidade de sólido na polpa, assim como a distribuição granulométrica que também desempenha uma importante função na filtragem.

Em um estudo, feito por Silva et al. (2015), diz ser possível fazer um teste prévio para a escolha do material que irá atuar como meio filtrante. O teste de filtragem em folha consiste em uma metodologia laboratorial que utiliza um suporte de filtragem padrão para ensaios. Neste teste, como resultado, é possível obter a quantificação de controle do nível de vácuo e porcentagem de sólidos da alimentação, lavagem da torta e temperatura da polpa.

Ainda no estudo de Silva et al. (2015), no teste mais satisfatório no quesito tempo de filtragem e umidade, foi visto um tempo de três minutos e umidade entre 15% e 22%.

Uma solução para amenizar o entupimento do meio filtrante é vista em um artigo de Turrer et al. (2010), que apresenta o uso de coagulantes para uma manutenção no desempenho do processo de filtragem. No processo de filtragem do minério de ferro é adicionado óxido de cálcio (cal) a polpa que ao misturar-se com a água gera uma coagulação das partículas sólidas, isto melhora o desempenho da filtragem, uma vez que os sólidos aglomerados contribuem para que a produtividade do processo esteja em patamares aceitáveis.

Há poucos estudos sobre o uso da visão computacional e inteligência artificial na indústria da mineração. O que pode ser visto é um estudo feito por Frick (2008), onde é mostrada a classificação de lascas de quartzo a partir do uso de visão computacional e redes neurais.

Diante da falta de estudos abordando a utilização de redes neurais e processamento de imagens para solucionar os problemas inerentes à filtragem de minério, a proposta deste trabalho tem como grande diferencial, a contribuição de uma solução usando visão computacional com redes neurais para classificar e mensurar a qualidade dos filtros, e a proposta de um indicador preventivo para o estado das telas dos filtros verticais desses equipamentos rotativos de secagem de minério.

3. DESENVOLVIMENTO PROPOSTO

O desenvolvimento do trabalho foi feito com base no fluxograma da Figura 2. Ele está dividido nas etapas de aquisição dos vídeos, separação e rotulação das imagens, criação do *Dataset*, a escolha dos modelos de rede neural, análise e comparação dos modelos e por fim, a criação de um indicador de qualidade para a formação da torta.

As subseções seguintes, irão detalhar os processos de desenvolvimento apresentados em cada bloco do fluxograma da Figura 2.

3.1 Aquisição das imagens

A aquisição das imagens foi realizada a partir de vídeos do equipamento que contém os filtros, disponibilizado de forma *offline* para este estudo. Foram utilizados três tipos de vídeos que mostram estados diferentes da formação das tortas nos filtros. Uma filmagem de uma torta de minério se formando por completo, que pode-se dizer ser uma torta de boa qualidade vista na Figura 3(c). Uma torta com uma formação mediana, a qual a manutenção ainda não é imediata, mas está próxima de ocorrer irregularidades, presente na Figura 3(a). E por fim, a Figura 3(b) representa uma

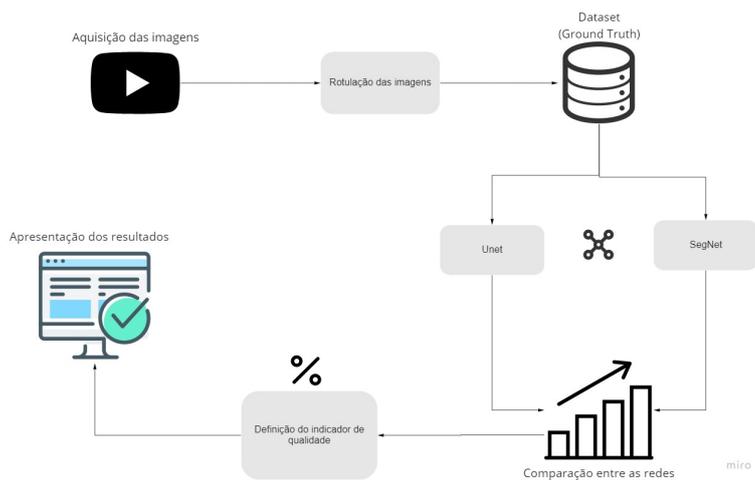


Figura 2. Fluxograma de desenvolvimento.

torta classificada como ruim, onde a formação da torta é vista em poucos pontos do meio filtrante.

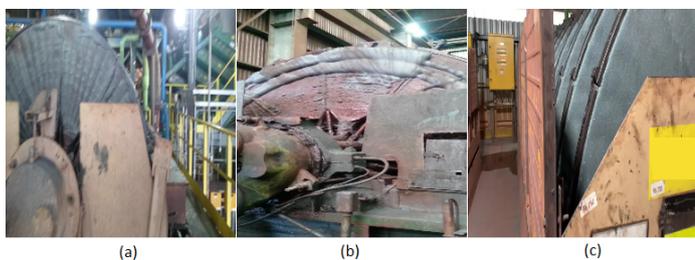


Figura 3. Imagens retiradas da filmagem dos filtros.

Para cada vídeo foi feita a extração das imagens em *frames*, obtendo figuras estáticas para a manipulação.

3.2 Rotulação

A etapa de rotulação das imagens foi feita utilizando um *software* chamado *Image Annotation Tool* (Kläser, 2010). Este *software* proporciona a criação de máscaras binárias para imagens, selecionando a área que deseja destacar. Na Figura 4 é possível ver alguns exemplos de máscaras obtidas pelo uso do *software*, construídas a partir das imagens originais.

3.3 Dataset

O *dataset* consiste no conjunto de dados que será usado para a realização do treinamento e validação das arquiteturas de redes neurais utilizadas. No caso deste estudo, ele é formado pela filmagem dos filtros verticais presentes na etapa de filtragem do minério.

A composição do *dataset* foi feita a partir da extração de imagens dos vídeos e as rotulações apresentadas anteriormente. Assim, foram separadas em conjuntos de entrada e saída, totalizando 120 amostras. Dessas, 1/3 foram obtidas para cada tipo de torta filmada, retiradas de diferentes trechos dos vídeos.

Além disso, imagens não rotuladas foram usadas para teste, de maneira a visualizar qualitativamente o comportamento das redes neurais aplicadas para imagens nunca vistas antes.

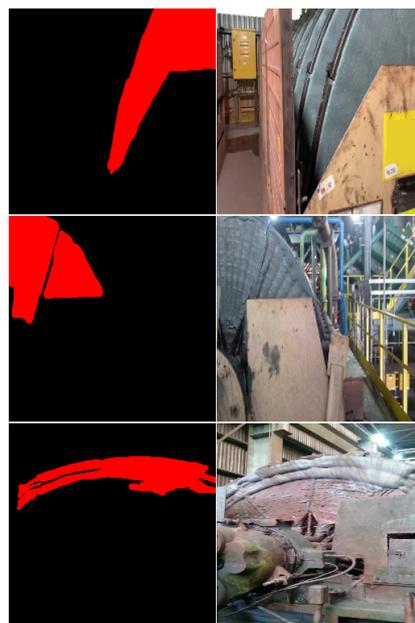


Figura 4. Exemplo de máscaras obtidas pelo *software Image Annotation Tool*.

3.4 Arquitetura de redes neurais aplicadas

Para este trabalho foram escolhidas duas arquiteturas semelhantes, porém cada uma com suas peculiaridades e eficiências diferentes. As arquiteturas utilizadas são a SegNet (Badrinarayanan et al., 2015) e a Unet (Ronneberger et al., 2015), redes neurais convolucionais, do inglês *Convolutional Neural Network* (CNN), do tipo *encoder/decoder*.

Unet: A arquitetura UNET é vista em Ronneberger et al. (2015) como uma aplicação de CNN para segmentação de imagens na Biomedicina. A ilustração da arquitetura do tipo Unet pode ser vista na Figura 5.

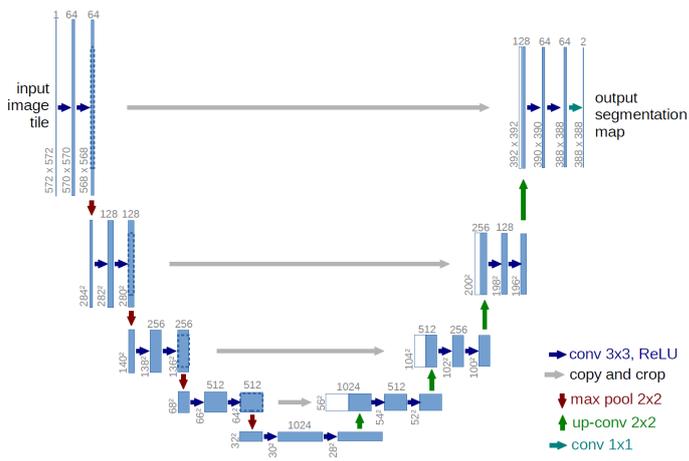


Figura 5. Arquitetura Unet. Fonte: Ronneberger et al. (2015).

A Unet possui um caminho, que consiste na contração (*encoder*) da imagem e outro caminho que é a expansão (*decoder*). A contração consiste na repetição de convoluções 3x3, tendo sua representação matemática vista na Equação 1, onde f é definida como a imagem original e g o filtro. Logo, $(f * g)$ caracteriza a resposta da convolução.

$$(f * g)(i, j) = \sum_a \sum_b f(a, b)g(i - a, j - b) \quad (1)$$

A convolução em imagens é feita com a correlação entre o *kernel* invertido e a imagem de entrada, multiplicando elemento por elemento, e o resultado de saída sendo a soma dessas multiplicações (Zhang et al., 2021).

Cada uma dessas convoluções é seguida por uma função de ativação que irá decidir se a saída da camada é relevante ou não para o andamento da rede. Usualmente é utilizada a função *ReLU*, representada pela Equação 2.

$$ReLU(x) = \max\{0, x\} \quad (2)$$

A função *ReLU*, em geral, irá ativar apenas os elementos positivos, descartando os negativos passando sua ativação para 0.

Após a convolução, uma operação de *pooling* máximo 2x2, que cria um agrupamento com as características extraídas, é aplicada. A operação irá selecionar o valor máximo entre a vizinhança, diminuindo a resolução.

Após fazer duas convoluções na etapa de contração, é feita uma convolução transposta 2x2. Ela funciona de forma que a saída esperada seja a inversa de uma convolução normal, assim a convolução aumenta a resolução da entrada, e esta é concatenada para a etapa de expansão.

O caminho de expansão possui a mesma convolução 3x3 seguida pela ativação *ReLU*. Por fim, na camada final, existe uma convolução 1x1 e uma ativação do tipo sigmoide (Equação 3), utilizada para mapear as 64 camadas e transformar os valores que estão na forma real para saídas de intervalos entre 0 e 1. Após isso, a saída apresenta uma imagem binária, representando a classe de cada pixel da imagem de entrada.

$$sigmoid(x) = \frac{1}{1 + exp(-x)} \quad (3)$$

SegNet: A arquitetura do tipo SegNet, vista na Figura 6, foi apresentada em Badrinarayanan et al. (2015). Ela consiste em 13 camadas convolucionais na etapa de codificação, sendo elas as mesmas 13 camadas utilizadas na arquitetura VGG16, apresentada por Simonyan and Zisserman (2014) e foi construída para a classificação de objetos.

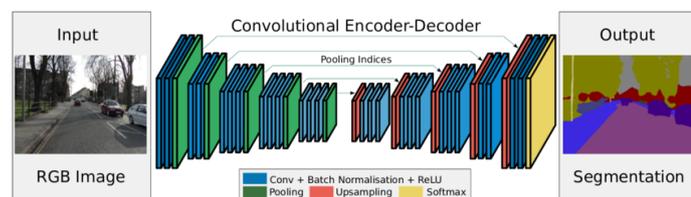


Figura 6. Arquitetura SegNet. Fonte: Badrinarayanan et al. (2015).

Cada camada de codificação da SegNet possui uma decodificação correspondente, sendo assim, a etapa de decodificação também possui 13 camadas. A codificação desta arquitetura possui uma convolução, seguida de uma normalização em lote, que consiste em normalizar a saída

subtraindo a média do lote e dividindo pelo desvio padrão do lote. O lote, que é passado como parâmetro da função de treinamento, é determinado pela quantidade de dados que irão ser utilizados a cada época. A função de normalização em lote é exemplificada na Equação 4, em que μ_B é a média e σ_B o desvio padrão.

$$BN(X) = \frac{x - \mu_B}{\sigma_B} \quad (4)$$

Assim, seguindo de uma ativação *ReLU*, e ao fim de cada camada é feito um *pooling* máximo 2x2. Pode-se reparar uma grande semelhança com a arquitetura apresentada anteriormente, Unet. O que difere as duas arquiteturas é a etapa de decodificação. Enquanto na Unet é transferido, a cada codificação, todo o mapa de característica para a camada de decodificação, processo que gera um maior custo de memória, na SegNet é apenas reutilizado os índices de agrupamento das camadas de codificação correspondente. Com isso, a decodificação é seguida de uma convolução, uma normalização em lote e uma ativação *ReLU*. Por fim, na última camada, a saída é adicionada a um classificador, chamado *soft-max*, representado na Equação 5, onde X é o valor do pixel na posição (i, j) , e trás a probabilidade de cada pixel para a quantidade de classes especificadas.

$$softmax(X)_{ij} = \frac{exp(X_{ij})}{\sum_k exp(X_{ik})} \quad (5)$$

3.5 Treinamento das redes

Ambos os modelos seguiram uma mesma linha de treinamento. O treinamento de uma rede neural consiste nas etapas de propagação e retro-propagação. Na propagação, a cada iteração é passado um determinado lote de dados pelas camadas definidas em cada modelo, o número de nós (neurônios) em cada camada varia conforme o tamanho da entrada.

Após terminada a propagação é feito o cálculo da perda, avaliando se a saída proporcionada pela rede obteve um desempenho desejável. Com isso, começa a etapa de retro-propagação que consiste em encontrar os valores desejados de saída e comparar com as saídas atuais, e com isso, por meio de uma função de otimização, atualizar os parâmetros de forma que o erro diminua.

A função de otimização utilizada no trabalho, que é também a mais comum nos treinamentos de redes neurais atuais, foi a função chamada Adam. A função Adam, proposta por Kingma and Ba (2014), tem o objetivo de minimizar a perda a cada época de treinamento. Segundo Ribeiro et al (2020), a função Adam calcula uma média da atualização anterior e uma variação da atualização atual, e então, faz uma normalização dos parâmetros de treinamento de forma que o resultado chegue próximo ao esperado.

3.6 Comparação de resultados

Para que seja possível comparar e avaliar o desempenho dos modelos, são passadas, durante o treinamento, métricas de avaliação para entender como está o comportamento dos modelos diante das diferentes épocas.

As arquiteturas receberam as mesmas métricas de avaliação em seus treinamentos, e também são avaliadas pela taxa de perda de cada treinamento, onde o intuito é ser minimizada a cada época. A função de perda consiste em calcular a porcentagem de pixel que não está na intersecção da imagem predita pela rede e a máscara real. Para esse cálculo é feita a divisão da intersecção das imagens pela união das duas.

As métricas usadas foram, a precisão e a *recall*. Começando pela precisão, ela é uma métrica determinada pela Equação 6, onde *VP* significa verdadeiros positivos, e *FP* falsos positivos. A precisão define quantos porcentos dos acertos da rede realmente são corretos.

$$precision = \frac{VP}{VP + FP} \quad (6)$$

A métrica *recall*, é definida na Equação 7, onde *FN* é falso negativo, e mostra a capacidade da rede de determinar se um resultado é verdadeiro.

$$recall = \frac{VP}{VP + FN} \quad (7)$$

3.7 Indicativo de qualidade

O indicativo de qualidade do preenchimento da torta no meio filtrante, levando em consideração a área preenchida, consiste em fazer uma comparação entre a máscara real, com o tamanho completo do meio filtrante, e o retorno da imagem classificada pelas redes neurais. Logo, o objetivo é encontrar/caracterizar uma porcentagem de preenchimento da torta no filtro.

Essa comparação é feita a partir da intersecção das duas imagens, fazendo a multiplicação dos elementos correspondentes de cada matriz de imagem binária, e assim, obtendo apenas elementos classificados como minério dentro da região da torta. Dessa forma, normalizamos a imagem de maneira que a classificação da rede se restrinja à apenas os pixels dentro da região do *ground truth* do meio filtrante.

Com isso, basta dividir a quantidade de pixels normalizados pela quantidade total estabelecida pelo *ground truth*, encontrando uma porcentagem de preenchimento da torta (Equação 8). A partir deste indicativo, pode-se então estabelecer regras de negócio quanto à qualidade da formação da torta em função dessa porcentagem de preenchimento. Assim, quando houver uma porcentagem abaixo do valor definido, esta referência indicaria uma possível necessidade de atenção no filtro.

$$IQ = \frac{TotalFiltro \cap AreaDaTorta}{TotalFiltro} \quad (8)$$

4. RESULTADOS

4.1 Ferramentas utilizadas

Para que seja possível obter e comparar os resultados encontrados pelas arquiteturas de rede, foi utilizado algoritmos desenvolvidos na linguagem *Python*. Para a visualização dos dados adquiridos, além das imagens de saída,

foi desenvolvido algoritmos para gerar gráficos, e mostrar o comportamento de cada rede.

Junto da linguagem *Python*, foram utilizadas algumas bibliotecas auxiliares. Para a parte das redes neurais, foi utilizada a biblioteca *TensorFlow*, que é uma biblioteca de código aberto composta de funções e modelos próprios para o aprendizado de máquina, e também aprendizado profundo. O *TensorFlow* nos permite utilizar sua própria função de treino, passando os parâmetros que desejamos visualizar e integrar à rede. Junto ao *TensorFlow*, utiliza-se a API Keras na implementação do modelo.

Vale ressaltar que o algoritmo utilizado foi baseado em algoritmos já existentes e disponibilizados na plataforma *GitHub*.

A leitura e manipulação das imagens é feita com o auxílio das bibliotecas, também de código aberto, *OpenCV* e *Numpy*, onde é possível manipular as imagens de forma que seja possível introduzi-las no modelo.

Para a elaboração dos gráficos e leitura dos dados adquiridos através dos testes e treinamentos, as bibliotecas *Pandas* e *Matplotlib*, foram de grande auxílio para o projeto. A primeira nos permite fazer a leitura de dados em tabelas e manipulá-los como *dataframes*, e a segunda possui funções para geração de gráficos.

Para a execução dos algoritmos usou-se a plataforma *Colaboratory*, uma ferramenta do Google, que nos permite rodar códigos em máquinas alocadas na nuvem.

4.2 Análise das Redes Neurais quanto aos resultados

Nesta seção os resultados obtidos para cada modelo de rede neural utilizada são apresentados. As duas arquiteturas foram utilizadas em suas estruturas comuns, alterando apenas os hiperparâmetros e suas métricas.

Para que seja possível entender o comportamento da rede de acordo com cada época do treinamento, foi utilizada 90% dos dados para a função de treinamento e 10% para teste. Além disso, para visualizar numericamente este comportamento, é usada uma função de *callbacks* da API Keras. As métricas e hiperparâmetros, com exceção do número de épocas, são passados para o modelo antes de chamar a função de treinamento.

Tabela 1. Hiperparâmetros das redes.

Parâmetros	Unet	SegNet
Otimização	Adam	Adam
Taxa de Aprendizado	0.00001	0.0001
<i>batch size</i>	2	2
Número de épocas	15	100
Métrica	<i>Precision</i>	<i>Precision</i>
Métrica 2	<i>Recall</i>	<i>Recall</i>

A Tabela 1 mostra os hiperparâmetros utilizados para treinamento das redes. Em ambas as redes utilizou-se o otimizador Adam. A taxa de aprendizado utilizada na arquitetura Unet é 0.00001, já na SegNet utilizou-se o valor de 0.0001. Esta taxa garante a convergência do modelo durante as épocas.

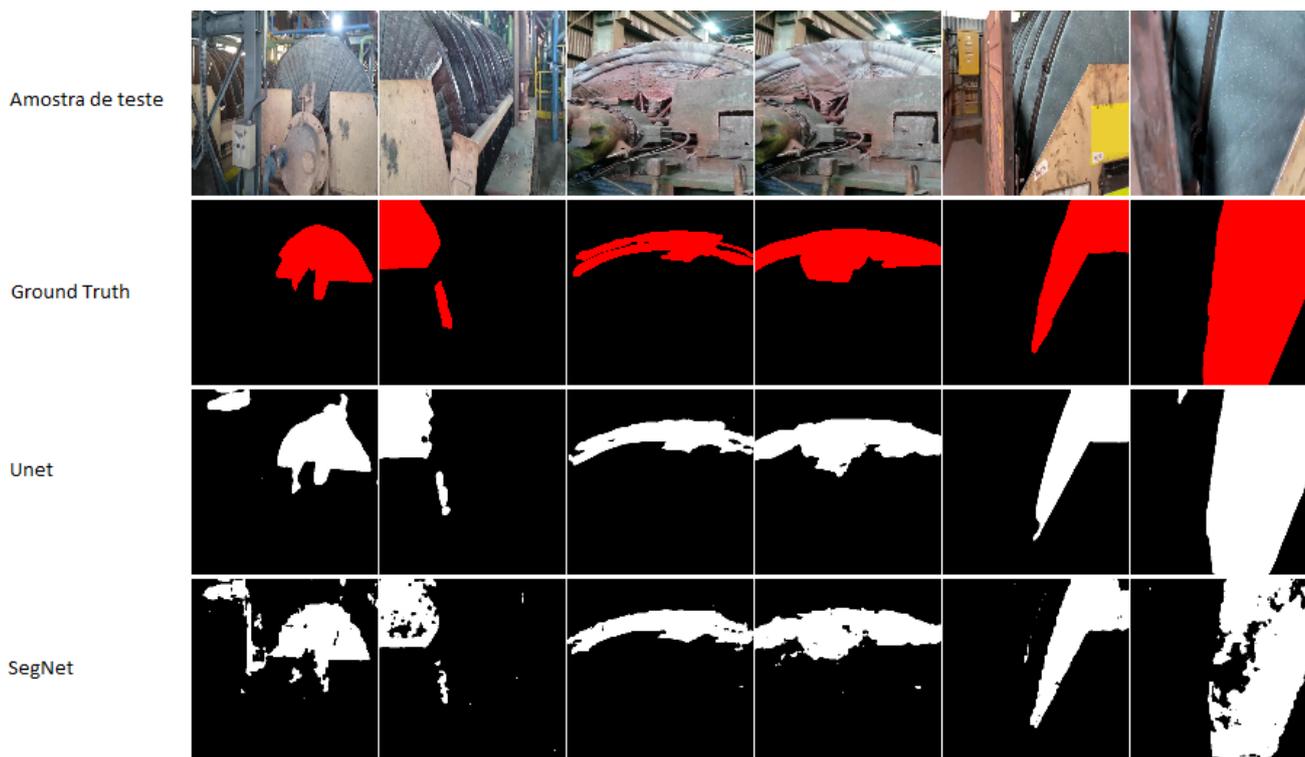


Figura 7. Comparação de resultados qualitativos obtidos pelas redes.

O parâmetro *batch size* divide o *dataset* em lotes, para que a cada época a rede treine com diferentes imagens. O valor 2 foi especificado para utilizar metade do *dataset* a cada época. Desta forma, o *batch size* atua dentro dos 90% das imagens definidas na fase de treino.

O parâmetro relacionado às épocas, é definido pela quantidade de vezes que a rede irá aplicar o treino, sendo cada época baseada no aprendizado obtido na época anterior. Na arquitetura Unet, 15 épocas foram suficientes para chegar a um resultado relevante. Já na SegNet foram necessárias 100 épocas, como pode ser visto na Figura 8.

Os parâmetros apresentados na Figura 8 são utilizados para modificar o comportamento da rede ao decorrer de cada época. As métricas *Precision*, *Recall* e a função de perda são utilizadas para quantificar o aprendizado da rede. A função de perda é baseada na interseção das imagens de saída do treinamento e o *ground truth* fornecido, e o objetivo é diminuí-la a cada época.

Os gráficos da Figura 8 mostram que a arquitetura Unet possui um aprendizado mais rápido, atingindo uma baixa taxa de perda e uma precisão considerável a partir da sexta época. Já a SegNet mostra um aprendizado mais lento, atingindo resultados considerados bons com precisão maior que 0.9 em torno da época de número 85. É importante dizer, que as imagens utilizadas para a validação das arquiteturas, não são as mesmas utilizadas para o treinamento, com isso, evita-se de gerar um *overfitting*, que seria um resultado precipitado e enviesado.

Os reflexos destes resultados são vistos nas imagens de saída de cada rede, utilizando as imagens de teste. A Figura 7, nos mostra o comparativo da imagem real, o *ground truth* da máscara, o resultado de saída da rede Unet

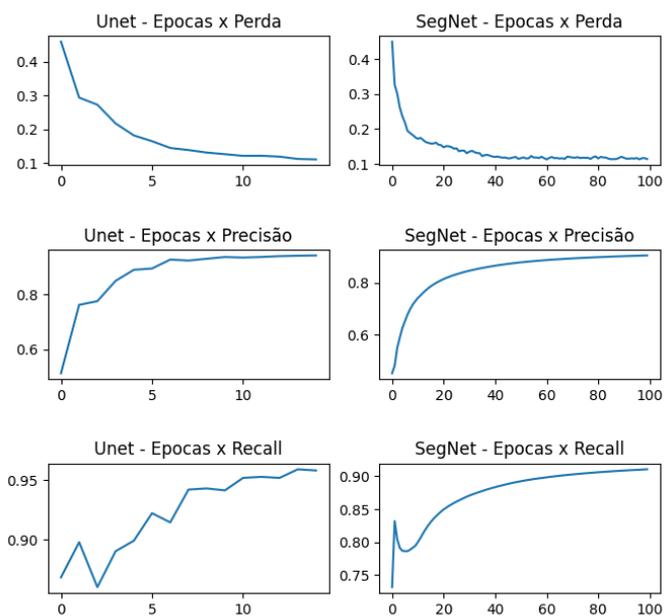


Figura 8. Comparação de métricas entre arquiteturas durante o treino.

e o resultado de saída da SegNet. Analisando a Figura 7, fica evidente que a arquitetura Unet chegou em resultados mais próximos do desejado, e com menos épocas em comparação a SegNet. Por outro lado é possível observar que a arquitetura SegNet, mesmo após 100 épocas, chega em um resultado com algumas falhas na segmentação, indicando a necessidade de uma maior quantidade de

épocas na fase de treinamento, o que torna o aprendizado ainda mais lento.

Em questão do tempo, a Unet rodou uma época em aproximadamente 265 milissegundos, enquanto a SegNet demorava 560 milissegundos, afirmando o seu maior tempo de aprendizagem.

Após os resultados obtidos pelas redes, foi gerado uma classificação para o nível de qualidade do preenchimento da torta, baseado na comparação da formação da torta em relação ao tamanho total do meio filtrante.

A Figura 9 apresenta a comparação para a classificação de uma torta de nível bom em (a), onde apresenta a imagem real da torta, a máscara representando o meio filtrante, e a segmentação das redes SegNet e Unet respectivamente. Em seguida, na Figura 9(b) é apresentado a classificação de uma torta intermediária. Por fim, a Figura 9(c) mostra uma torta ruim, com necessidade de manutenção.

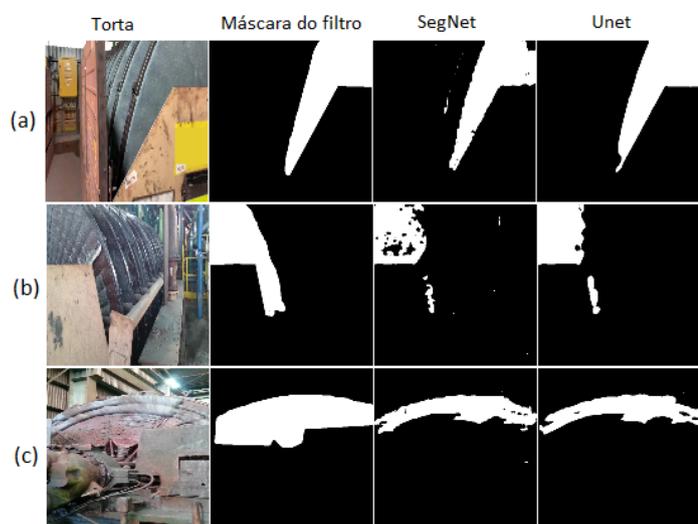


Figura 9. Comparação de qualidade entre o tamanho do meio filtrante e a segmentação da torta encontrada

A Tabela 2 apresenta o resultado do indicador de qualidade (*IQ*) para cada arquitetura de rede, de acordo com cada bloco de imagens apresentada pela Figura 9. Baseado nos resultados obtidos, um possível intervalo de classificação seria: valores maiores que 80% indicam uma boa formação de torta, valores entre 80% e 60% caracteriza-se uma torta intermediária (em fase de degradação) e valores abaixo de 60% seria um indicativo de má formação da torta. Neste ponto do processo, o ideal seria uma avaliação de um especialista que, baseado nos resultados qualitativos da Figura 9 e dos resultados quantitativos da Tabela 2, definiria o intervalo de classificação ideal para a aplicação.

Tabela 2. Indicativo de qualidade da torta

Figuras	Unet	SegNet
Figura 9(a)	92.28%	88.37%
Figura 9(b)	70.13%	62.64%
Figura 9(c)	49.86%	46.48%

5. CONCLUSÃO

Os resultados apresentados mostram que é possível utilizar as tecnologias de visão computacional e inteligência arti-

ficial para criar um indicador de qualidade das tortas no processo de filtragem de filtros verticais a vácuo. Tal tecnologia se mostra promissora no estudo, e é capaz de trazer uma previsibilidade para o mal preenchimento da torta, fazendo com que o problema seja evitado, minimizando a paralisação da produção.

Também foi possível avaliar o desempenho de duas arquiteturas de redes neurais aplicadas no problema, onde ficou demonstrado que a escolha da arquitetura de rede neural influencia diretamente nos resultados do experimento.

Para estudos e implementações futuras, pretende-se utilizar outras abordagens de classificadores a fim de entender o melhor comportamento da inteligência artificial diante do problema. Seria interessante, buscar outros dados como a espessura da torta, vazão mássica, pressão de sopro, pressão do vácuo e a umidade da torta, com o intuito de relacioná-los junto ao indicador de qualidade de preenchimento da torta no filtro, com o objetivo de obter um resultado mais preciso e prático.

REFERÊNCIAS

- Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561. URL <http://arxiv.org/abs/1511.00561>.
- Cesário Neto, E.D. (2020). *Os impactos da Indústria 4.0 na mineração*. Ph.D. thesis, Trabalho de Conclusão de Curso (Graduação em Ciências Econômicas).
- Frick, M.A.D. (2008). *Caracterização de minério de ferro por visão computacional*. Ph.D. thesis, Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Maria.
- Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. doi:10.48550/ARXIV.1412.6980. URL <https://arxiv.org/abs/1412.6980>.
- Kläser, A. (2010). Image annotation tool with image masks. URL https://lear.inrialpes.fr/people/klaeser/software_image_annotation.
- Luz, A.B.d. and Lins, F.A.F. (2018). Introdução ao tratamento de minérios. CETEM/MCTIC.
- Oliveira, M.L., Luz, J.A., and Lacerda, C.M. (2004). Espessamento e filtragem. URL https://files.cercomp.ufg.br/weby/up/596/o/apostila_ufop.pdf.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W.M. Wells, and A.F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, 234–241. Springer International Publishing, Cham.
- Silva, A.M.e., Paniago, E.B., Faria, D.L.A., Wendling, F., and Otaviano, M.M. (2015). CaracterizaÇão do processo de degradaÇão de uma fibra de poliamida utilizada como tecido filtrante na filtragem de uma polpa de minÉrio de ferro. In *Tecnologia em Metalurgia, Materiais e Mineração*, volume 3, 28–33. ABM. doi: <http://dx.doi.org/10.4322/tmm.00303006>.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. doi:10.48550/ARXIV.1409.1556. URL <https://arxiv.org/abs/1409.1556>.

- Soares, P.H.V., Jesus, W., José, R.F.d.S., Gomes, L., Aguiar, A.B., Rodrigues, V.J.P., and Satyro, J.C.d.S. (2017). *REDUÇÃO DA UMIDADE NO PELLET FEED DA USINA CONCEIÇÃO I*, volume 1, 124–135. Blucher Preceeding, 18^o edition.
- Turrer, H.D.G., Junior, A.M.A., Doellinger, T.M., Zuchi, M.A., and Peres, A.E.C. (2010). Uso de coagulantes para manutenção do desempenho da filtração de minério de ferro. In *Tecnologia em Metalurgia, Materiais e Mineração*, volume 7, 42–48. ABM. doi:<http://dx.doi.org/10.4322/tmm.00701008>.
- Zhang, A., Lipton, Z.C., Li, M., and Smola, A.J. (2021). Dive into deep learning. doi:10.48550/ARXIV.2106.11342. URL <https://arxiv.org/abs/2106.11342>.