UNIVERSIDADE CATÓLICA DOM BOSCO
Centro de Ciências Exatas e da Terra
Engenharias de Controle e Automação, Computação, Elétrica e Mecânica

HISTOGRAMA DE SUPERPIXELS PARA APLICAÇÕES EM PALINOLOGIA FORENSE

Alexandre Fernandes Cese
David Augusto Guimarães

Campo Grande - MS

Dezembro, 2017

HISTOGRAMA DE SUPERPIXELS PARA APLICAÇÕES EM PALINOLOGIA FORENSE

Alexandre Fernandes Cese
David Augusto Guimarães

Projeto de Graduação submetido à Coordenação do curso de Bacharelado em Engenharia de Computação da Universidade Católica Dom Bosco como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Hemerson Pistori

Campo Grande - MS

Dezembro, 2017

“(...) Failure is not an option.”

Gene Kranz.

Dedicatória

A Deus, que em sua infinita sabedoria guia meus caminhos me proporcionando saúde, serenidade e disposição para enfrentar todas as etapas desta árdua caminhada.

A minha mãe Dalva que com seu amor infinito e apoio incondicional é responsável por minha base pessoal e educacional. Sem ela, esta conquista não seria possível.

Ao meu pai Marcos que como minha mãe, foi peça fundamental para minha formação como pessoa e como profissional. Pai, agradeço profundamente todo o apoio incondicional ao longo de todos esses anos. Esta conquista também é sua!

A minha melhor amiga, minha irmã Vanessa, que sempre acreditou que eu seria capaz de realizar meus sonhos mais ousados e sempre me incentivou.

A minha irmã Viviane. Vi, não levo o mérito de ser a primeira pessoa da família a conquistar um diploma, pois reconheço que este mérito seria seu caso ainda estivesse conosco.

David Augusto Guimarães

AGRADECIMENTOS

A minha família, que sempre me apoiou em todos os momentos de minha vida. Meu muito obrigado a minha mãe Dalva, meu pai Marcos, minha irmã Vanessa, meu cunhado Valteir, meu tio Marte, minha tia Luciana, minha avó Maria, minha prima Louyne, meu primo Hemerson, bem como meus sobrinhos Vítor, Luiz Henrique, Anthony Vinícius e Amanda. Vocês têm sido minha base desde o meu nascimento.

Ao meu orientador, o Doutor Hemerson Pistori pela paciência, atenção, compreensão e sabedoria dispensadas a mim, não somente durante o período de elaboração deste trabalho, mas também durante todo o período em que estivemos trabalhando juntos ao longo de minha graduação.

Ao meu parceiro de trabalho Alexandre Cese que foi indispensável para o sucesso deste trabalho.

Meus mais sinceros agradecimentos aos Professores Kléber Padovani e Lia Nara por todo conhecimento e apoio transmitidos ao longo das disciplinas em que estivemos juntos. Agradeço também ao Professor Marcos Alves por todos os conselhos, conversas, paciência, além de todo conhecimento transmitido ao longo de todos os anos de minha graduação. Ao Professor Alexsandro Monteiro Carneiro por sempre ter apostado suas fichas em meu potencial, proporcionando diversas experiências que foram importantes para minha formação, como por exemplo, a possibilidade de trabalhar com projetos de pesquisa na universidade. Ao Professor Mauro Conti pela amizade, por acreditar em meu potencial, e por todo

conhecimento transmitido dentro e fora de sala de aula. Todos vocês foram fontes de inspiração durante todos esses anos.

Aos amigos que no convívio tornaram suportáveis as horas mais difíceis e mais felizes os momentos de vitória. Só tenho a agradecer por ter pessoas tão especiais em minha vida. Dentre todas os amigos que estiveram presentes ao longo desta trajetória de 7 anos, gostaria de agradecer especialmente a Thales Beraldo, André Fontolan, Leonardo Nazário, Edson Pinheiro, Daniela Milreu, Rainara Araújo, Patrícia Guedes, João Máximo, Daniela Duarte, Guidorjan Grubel, Sérgio Sousa, Leonardo Sousa, Leonardo Bellei, e Guilherme Henrique Oliveira. Tenho certeza que vivemos momentos maravilhosos juntos os quais me recordarei para sempre.

A minha namorada Amanda Verão que além de contribuir para as ilustrações empregadas neste trabalho, esteve comigo durante este último ano me apoiando nos momentos mais difíceis, transmitindo-me todo amor e carinho necessários para me convencer de que daria tudo certo no final.

David Augusto Guimarães

AGRADECIMENTOS

A minha família por sempre se fazer presente e me apoiar durante todo o processo de elaboração deste trabalho.

Ao meu orientador Hemerson Pistori pela paciência e orientação durante o período de elaboração deste trabalho.

Ao meu parceiro de trabalho David Guimarães por toda a ajuda.

Ao colega Alan Taranti pelas contribuições dadas ao nosso projeto.

Alexandre Fernandes Cese

HISTOGRAMA DE SUPERPIXELS PARA APLICAÇÕES EM PALINOLOGIA FORENSE

RESUMO

O estudo da classificação e estrutura de grãos de pólen (palinologia) é de fundamental importância para diversas aplicações nas áreas da biologia e apicultura. Por exemplo, através deste estudo é possível determinar a procedência e a qualidade do mel, assegurando a saúde das pessoas que o ingerem e evitando fraudes. Uma área correlata oriunda da palinologia é a palinologia forense, que trata da análise de grãos de pólen em investigações cíveis e criminais. A coleta e análise destes grãos em cenas de crime, por exemplo, podem trazer informações importantes para os peritos. No entanto, não se trata de uma tarefa trivial. Dentre os métodos de análise de grãos de pólen podemos destacar a análise microscópica feita por seres humanos, que apresenta limitações por conta de fatores biológicos (como por exemplo, o cansaço físico) e requer um profissional especializado para realizar a tarefa. O projeto PALINOVIC busca automatizar o processo de detecção, classificação e quantificação de grãos de pólen. Isto é feito através de técnicas e algoritmos de visão computacional e em alguns casos inteligência artificial, dentre as quais a técnica SLIC referente a *superpixels* e aprendizagem automática (supervisionada) foram escolhidas para serem aplicadas neste trabalho. Como este trabalho retrata o que já foi produzido anteriormente pelo projeto PALINOVIC, temos como objetivo aplicar novas técnicas para melhorar os resultados do *software* já existente. Isto será possível através de um estudo detalhado das técnicas de aprendizagem automática, K-médias (ou "*K-means*" como é chamado em Inglês) e *superpixels*, implementação dos algoritmos SLIC e K-médias, validação do módulo implementado, integração com sistema já existente, e por fim, análise e discussão dos resultados.

Palavras-Chave: palinologia; palinologia forense; *superpixels*; aprendizagem automática; visão computacional.

Lista de Figuras

Figura 1: Estrutura do órgão masculino de flores. 14
Figura 2: Ilustração de paredes interna (intina) e externa (exina) de um grão de pólen. 15
Figura 3: Exemplo de grão de pólen do tipo Anadenanthera, capturado em microscópio LCD Micro Bresser, sob objetiva de 40X. 15
Figura 4: Ilustração de segmentação de imagens utilizando a técnica de superpixels através de uma implementação do algoritmo SLIC. Na imagem, o algoritmo é executado no software PYNOVISAO, desenvolvido pelo grupo INOVISAO. 22
Figura 5: Árvore de decisão utilizada para concessão de bolsas de estudo. 27
Figura 6: Ilustração de uma Random Forest. 29
Figura 7: MLP. 30
Figura 8: Sigmoide e funções de custo. 31
Figura 17 – Pólen segmentado em superpixels 37
Figura 18: Exemplo de uma imagem com o seu histograma de superpixel 38
Figura 9: Algoritmo para extrair atributos de cada superpixel da imagem. 39
Figura 10: Primeira parte do algoritmo para achar os histogramas por imagem na pasta. 40
Figura 11: Segunda parte do algoritmo para achar os histogramas por imagem na pasta. 40
Figura 19: Fluxo de trabalho do histograma de superpixels em uma aplicação real 41
Figura 12: Matriz de confusão do SVM com k=85 aplicado nos dados de teste. 46
Figura 13: Imagens do pólen acoita cavalo (1). 49
Figura 14: Imagens da cordia trichotoma (2). 49
Figura 15: Imagens da myracroduon urundeuva (3). 49
Figura 16: BoxPlot dos métodos de classificação. 50

Lista de Tabelas

Tabela 1. Medidas-F de k no intervalo entre 10 e 100 aplicando validação cruzada nos dados de treinamento normalizados. 44
Tabela 2. Medidas-F de k no intervalo entre 10 e 100 aplicando validação cruzada nos dados de treinamento não-normalizados. 45
Tabela 3. Medidas-F dos resultados do SVM com k=85 aplicado nos dados de teste normalizados. 45
Tabela 4. PCC de todas as classes de pólens testadas. 47
Tabela 5: Nível de significância entre os métodos de classificação. 51

SUMÁRIO

1. INTRODUÇÃO **13**

2. OBJETIVOS **17**

2.1. Objetivos Gerais 18

2.2. Objetivos Específicos 18

3. FUNDAMENTAÇÃO TEÓRICA **18**

3.1. Palinologia Forense 19

3.2. Visão Computacional 20

3.2.1. Segmentação de Imagens 21

3.2.2. Segmentação em Superpixels 22

3.2.3. SLIC 23

3.2.4. Normalização dos atributos de superpixel antes de aplicar o k-means 24

3.2.5. Normalização dos histogramas para a classificação das imagens 24

3.3. Aprendizagem de Máquina 24

3.3.1. Aprendizagem Supervisionada 25

3.3.1.1. SVM (SMO) 26

3.3.1.2. Árvore de Decisão 26

3.3.1.3. Random Forest 28

3.3.1.4. MLP 29

3.3.1.5. Regressão Logística 30

3.3.2. Aprendizagem Não Supervisionada 31

3.3.2.1. Algoritmo K-médias 32

3.3.2.2. Mini Batch K-means 33

3.3.3. Aprendizagem Semi-Supervisionada 34

3.4. Métricas de Classificação 34

3.4.1. PCC 34

3.4.2. Medida-F 35

3.4.3. Matrizes de Confusão 35

3.5. Técnicas de Amostragem 36

3.5.1. Validação Cruzada 36

4. HISTOGRAMA DE SUPERPIXELS **36**

5. METODOLOGIA **42**

5.1. Estudo Detalhado da Técnica de Segmentação por Superpixels Baseada no Algoritmo SLIC 42

5.2. Implementação do Novo Extrator de Atributos “Histograma de Superpixels” 42

5.3. Validação do Módulo Implementado 43
5.4. Integração com Sistema Já Existente (PYNOVISAO) 43
6. RESULTADOS E DISCUSSÃO 44
6.1. Estatísticas dos Métodos de Classificação 50
7. CONSIDERAÇÕES FINAIS 52
8. REFERÊNCIAS BIBLIOGRÁFICAS 53

1. INTRODUÇÃO

A palinologia é a ciência que estuda pólens. O estudo de pólens exerce um papel importante em diversas áreas, como, por exemplo, na apicultura, uma vez que a análise polínica torna possível determinar a procedência de produtos apícolas. Por exemplo, uma vez que se torna possível determinar a procedência do mel de abelha (tradicional produto apícola), torna-se possível também garantir a qualidade do produto.

Para que a análise dos pólens seja possível, faz-se necessário antes realizar a sua coleta. A coleta dos pólens acontece em locais específicos, de acordo com o problema em questão. Exemplos da aplicação de pólens e seus locais de coleta são os estudos arqueológicos e forenses, onde a coleta no primeiro exemplo acontece em fósseis e no segundo exemplo, em cadáveres e cenas de crime. Da necessidade de estudos de pólens em contextos periciais surgiu uma área oriunda da palinologia: a palinologia forense, que é alvo principal deste trabalho.

A palinologia forense consiste no estudo de pólens em contextos periciais, de forma que as informações coletadas através dos pólens sejam utilizadas na solução de crimes. Esta área surgiu em meados da década de 1950 com seus primeiros registros históricos em 1959 (BRYANT, 2009), quando constatou-se que seria possível vincular pessoas e objetos a lugares específicos através destas partículas microscópicas. Trata-se de uma georreferência: cada região geográfica possui plantas com pólens específicos. Desta forma, a palinologia forense, apesar de ainda não ser largamente utilizada, tornou-se uma importante aliada na solução de crimes sempre que pólens podem ser coletados em cenas de crimes.

Gonçalves (2015) descreve que o grão de pólen faz parte do órgão masculino de flores, que é composto por filete, antera e androceu, bem como carrega o material gênico que é responsável pela fecundação das flores. A Figura 1 ilustra esta estrutura.

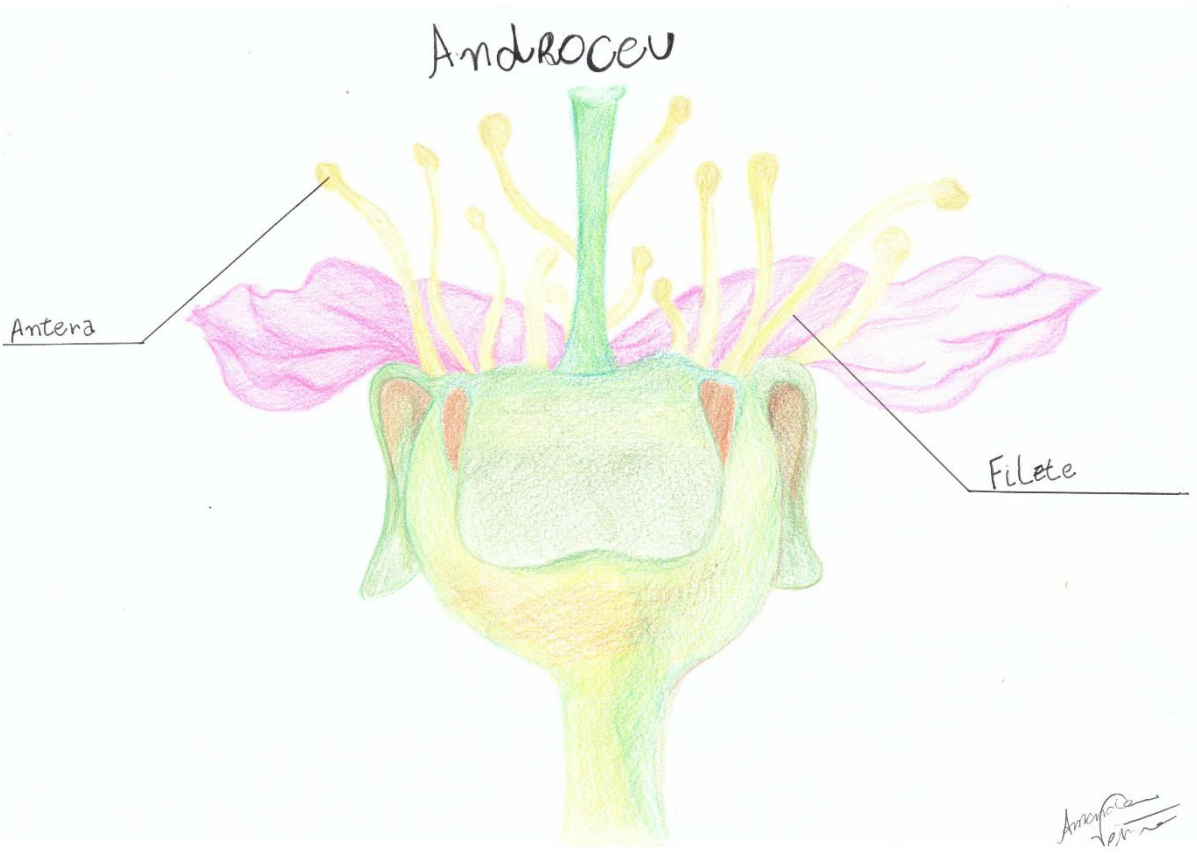

Figura 1: Estrutura do órgão masculino de flores.

Os pólens são compostos de basicamente duas estruturas: as paredes interna e externa, também chamadas de intina e exina, respectivamente. Tais estruturas apresentam certas aberturas ou fendas, que são chamadas de poros e permitem que o material gênico passe para que a fecundação seja feita. Tais aberturas nos pólens podem ser utilizadas como informações para que os grãos sejam identificados e categorizados, uma vez que estas características variam de acordo com a espécie do pólen.

A exina tem um papel crucial no processo de análise e identificação do pólen, porque é nesta estrutura em que é possível visualizar os poros dentre outras características, que tornam possível a identificação do mesmo (GONÇALVES, 2015). A Figura 2 ilustra essas duas estruturas do grão de pólen. A Figura 3 traz um exemplo de grão de pólen do tipo Anadenanthera, capturado em microscópio LCD Micro Bresser, sob objetiva de 40X.

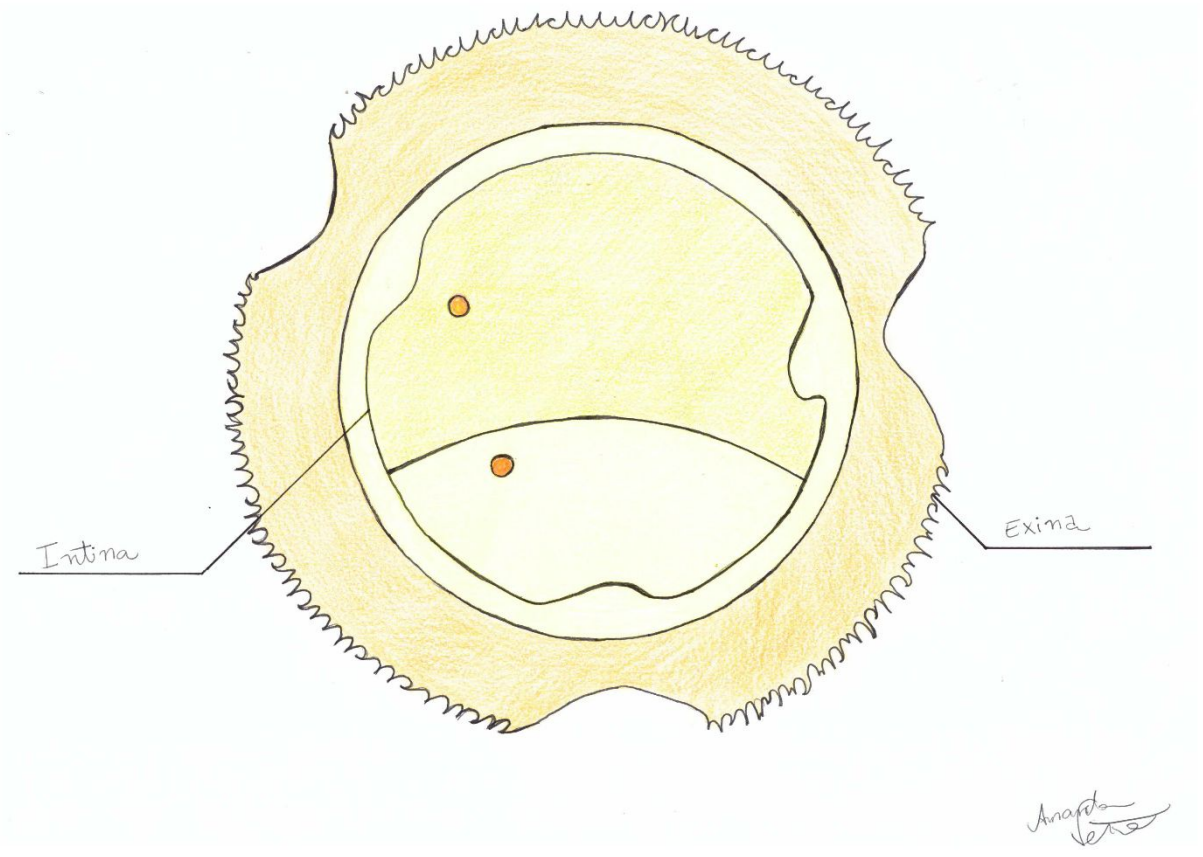

Figura 2: Ilustração de paredes interna (intina) e externa (exina) de um grão de pólen.

Figura 3: Exemplo de grão de pólen do tipo Anadenanthera, capturado em microscópio LCD Micro Bresser, sob objetiva de 40X.

No entanto, a determinação dos grãos de pólen não se trata de uma tarefa trivial, uma vez que a exina é muito parecida em determinadas espécies. Por este motivo, em alguns casos, faz-se necessário classificá-los em tipos polínicos, que seria uma espécie de categoria de pólens. Sendo assim, tal tarefa é desenvolvida por profissionais da área capacitados a fazer tal identificação visual, utilizando ferramentas como um microscópio, por exemplo.

A classificação de grãos de pólen por análise microscópica através da visão humana, contudo, apresenta limitações referentes a fatores biológicos (como por exemplo, o cansaço físico) que comprometem o processo de classificação ao longo do tempo. Isto ocorre devido a monotonia e repetitividade desta atividade que, após ser executada por várias horas por uma pessoa, pode levar a ocorrência de erros.

O Projeto PALINOVIC tem como finalidade automatizar o processo de classificação e de quantificação de grãos de pólens, para a obtenção de resultados mais confiáveis, precisos e rápidos. Tal automação é feita por meio de algoritmos de visão computacional que tratam imagens microscópicas. Dentre as diversas técnicas do campo da visão computacional e inteligência artificial, foram selecionadas para este trabalho, a técnica de *superpixels*, K-médias e aprendizagem automática (supervisionada), respectivamente. Este trabalho retrata o que já foi produzido pelo projeto PALINOVIC, aplicando novas técnicas para que, assim, seja possível tratar o problema de maneiras diferentes e mais eficazes, tornando assim, os resultados mais precisos e confiáveis.

A técnica de segmentação por *superpixels* tem sido aplicada em trabalhos correlatos no grupo INOVISAO[1] com resultados satisfatórios. O principal objetivo da utilização da técnica de segmentação de imagens por *superpixels* é a possibilidade de categorização dos objetos da imagem através de sua segmentação. Uma vez que a imagem é segmentada em *superpixels*, o algoritmo K-médias fica responsável por essa categorização por meio da "aglomeração" dos *superpixels* em um número *k* de classes (ou categorias, como temos dito até o momento). Os algoritmos SMO (*Sequential Minimal Optimization*) de SVM (Máquinas de Vetores de Suporte ou "*Support Vector Machines*" em Inglês), Árvore de Decisão, *Random Forest*, MLP (*Multi-Layer Perceptron"* ou "*Perceptron* Multicamadas" em Português) e Regressão Logística de aprendizagem de máquina são utilizados no treinamento do *software*

[1] Grupo de Pesquisa em Visão Computacional da Universidade Católica Dom Bosco.

para que o mesmo seja capaz de fazer as classificações dos pólens nas imagens de forma automática.

2. OBJETIVOS

2.1. Objetivos Gerais

Desenvolver um *software* de visão computacional baseado em um novo algoritmo proposto neste trabalho que seja capaz de realizar, de forma rápida e eficiente, a classificação de grãos de pólen por meio da análise de imagens microscópicas, tornando seus resultados mais precisos e confiáveis, melhorando, então, os resultados de um *software* já existente do projeto PALINOVIC chamado PYNOVISAO.

2.2. Objetivos Específicos

Para atingir o objetivo geral definido na seção 2.1, foram estabelecidos os seguintes objetivos específicos:

- Estudo Detalhado da Técnica de Segmentação por *Superpixels* Baseada no Algoritmo SLIC;
- Implementação do Novo Extrator de Atributos "Histograma de *Superpixels*";
- Validação do Módulo Implementado;
- Integração com Sistema Já Existente (PYNOVISAO);
- Análise e Discussão dos Resultados.

3. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentadas as técnicas e algoritmos que fundamentam este trabalho.

3.1. Palinologia Forense

Palinologia é a ciência que estuda pólens, que são partículas microscópicas que estão no ar e aderem-se a roupa e a pele a todo o momento (WRAY, 2016). É utilizada como ferramenta em diversos contextos diferentes, dentre os quais podemos destacar investigações de crimes e estudos arqueológicos para determinar a idade de fósseis. Independente do contexto em que esta ciência é inserida, ela envolve a coleta de pólens, uma vez que é necessário primeiro coletar os pólens para que o mesmos possam ser analisados posteriormente.

É interessante destacar que os locais de coleta de pólens está diretamente relacionado ao contexto em que o problema está inserido. Seguindo os exemplos citados anteriormente, se a análise polínica será utilizada para determinar a idade de fósseis, o local de coleta dos pólens será nos fósseis em questão. Ainda, se a análise polínica está sendo utilizada em contextos periciais, a coleta dos pólens será feita em cenas de crime ou em cadáveres.

Já a palinologia forense é uma área oriunda da palinologia. Existem aproximadamente de 7 a 8 milhões de espécies de plantas, cada qual com um tipo de pólen específico. Tais plantas podem ser polinizadas de formas diferentes, como por exemplo através do vento ou a partir de outros insetos. Vale lembrar que um número maior de espécies são polinizadas através do vento e que neste caso os pólens conseguem atingir distâncias maiores do que nos casos em que são transportados por insetos (BRYANT, 2009). Por este motivo foi dito anteriormente que trata-se de partículas microscópicas que estão no ar e aderem-se a pele e as roupas a todo momento.

Estes fatos aliados ao fato de que espécies específicas de plantas (e consequentemente pólens) são predominantes em determinadas regiões geográficas fizeram com que surgisse o campo da palinologia forense dentro da área da palinologia em si, uma vez que torna-se possível referenciar pessoas e objetos a determinadas localidades geográficas, a

partir dos pólens daquela região. É interessante notar que o período do ano em que o crime ocorreu pode ser referenciado.

Este tipo de investigação ocorre há anos na Europa, Austrália e está começando a ter popularidade nos EUA. O grande problema do processo é a classificação dos pólens, já que há milhares de espécies de pólens e muitos são parecidos, além do fato de que, por investigação, há geralmente centenas de pólens e verificar um a um é um trabalho demorado e árduo (WRAY, 2016). É daí que surge a necessidade de se fazer programas de computador que têm a possibilidade de serem mais eficientes do que humanos e podem ser mais rápidos também.

3.2. Visão Computacional

Visão computacional é a área da ciência da computação que busca emular os mecanismos de visão de um ser humano. Para humanos e animais, o reconhecimento de objetos, padrões e pessoas nos mais variados ambientes e distâncias é uma tarefa muitas vezes considerada como trivial, uma vez que acontece de forma natural. Por este motivo, a visão computacional muitas vezes é tratada erroneamente como uma área trivial.

No entanto, reproduzir computacionalmente tais mecanismos da visão humana pode ser uma tarefa demasiadamente complicada, uma vez que para tal, a visão computacional baseia-se em métodos matemáticos muitas vezes avançados. Tendo isto em mente, é correto afirmar também que a visão computacional busca a integração de vários dos processos utilizados na visão humana, que são descritos matematicamente e executados computacionalmente (BALLARD & BROWN, 1982).

Com o avanço tecnológico, cada vez mais aplicações nas mais diversas áreas tem sido passíveis de automação a partir de *softwares* de visão computacional, o que tem contribuído muito para o avanço de pesquisas na área. Dentre exemplos de aplicações, podemos citar: sistemas de navegação para carros autônomos, *softwares* de reconhecimento facial para redes sociais e sistemas de vigilância, contador de alevinos automático, entre outros.

Como trata-se de *softwares* muitas vezes complexos, um sistema de visão computacional conta com sub etapas previamente definidas. São estas: aquisição da imagem,

pré-processamento, segmentação, extração de atributos e processamento em alto nível. Na primeira etapa, a imagem ou o conjunto de imagens a serem trabalhadas pelo sistema é capturada através de sensores e câmeras. Na segunda etapa, são realizadas algumas melhorias (que neste contexto chamamos de filtros) na imagem afim de tornar mais fácil os trabalhos realizados nas etapas seguintes. Um clássico exemplo de filtros é a suavização nas bordas da imagem (PISTORI, 2013).

A terceira etapa consiste em isolar na imagem, por meio de segmentação, somente as partes que desejamos trabalhar, como uma coleção de objetos, por exemplo. É importante notar que o fato de reduzir a imagem em estruturas menores e homogêneas que concentram somente as partes interessantes para a análise torna o processo em si mais rápido, demandando consequentemente menor custo computacional. A etapa seguinte é responsável por extrair atributos das imagens segmentadas com o objetivo de classificá-las, posteriormente.

Por último, todas as informações reunidas, até então, são aplicadas em algum problema específico. No entanto, é fundamental destacar que algumas dessas etapas de um sistema de visão computacional podem ser descartadas em problemas específicos. Por exemplo, no caso de um problema envolvendo redes neurais convolucionais, a etapa de extração de atributos não é necessária. Ainda, o pré-processamento pode não ser estritamente necessário em alguns casos. Por exemplo, se o processo de captura de uma determinada imagem não sofreu perturbações ou ruídos, a aplicação de filtros na etapa de pré-processamento pode ser descartada.

3.2.1. Segmentação de Imagens

Como dito anteriormente, o processo de segmentação de uma imagem é uma das sub etapas de um sistema de visão computacional e tem um papel importante no sistema como um todo, uma vez que a restrição e enfoque para as partes da imagem, as quais desejamos trabalhar, faz com que o esforço computacional no processo de análise da imagem seja menor, tornando o processo todo mais rápido e eficiente.

Este enfoque nas partes mais interessantes para uma dada aplicação específica torna-se possível a partir das regiões menores que compõem a imagem, também chamados de

segmentos. Ainda, cada segmento da imagem é único, o que implica dizer que os segmentos não se sobrepõem e cada *pixel* da imagem pertence a somente um de seus segmentos (SALDANHA, 2008).

3.2.2. Segmentação em *Superpixels*

Superpixels são agrupamentos de *pixels*, sendo que os *pixels* de uma imagem serão agrupados de acordo com semelhanças pré-estabelecidas. Como discutido anteriormente, técnicas de segmentação de imagens tem como objetivo principal gerar uma imagem final divididas em regiões menores, chamadas de segmentos. Sendo assim, *superpixels* consiste em uma técnica de segmentação cujo objetivo é a representação da referida imagem em um grupo menor de *pixels*, considerando que os *pixels* da imagem serão aglomerados em um segmento que é o *superpixel* em si (LINARES, 2013).

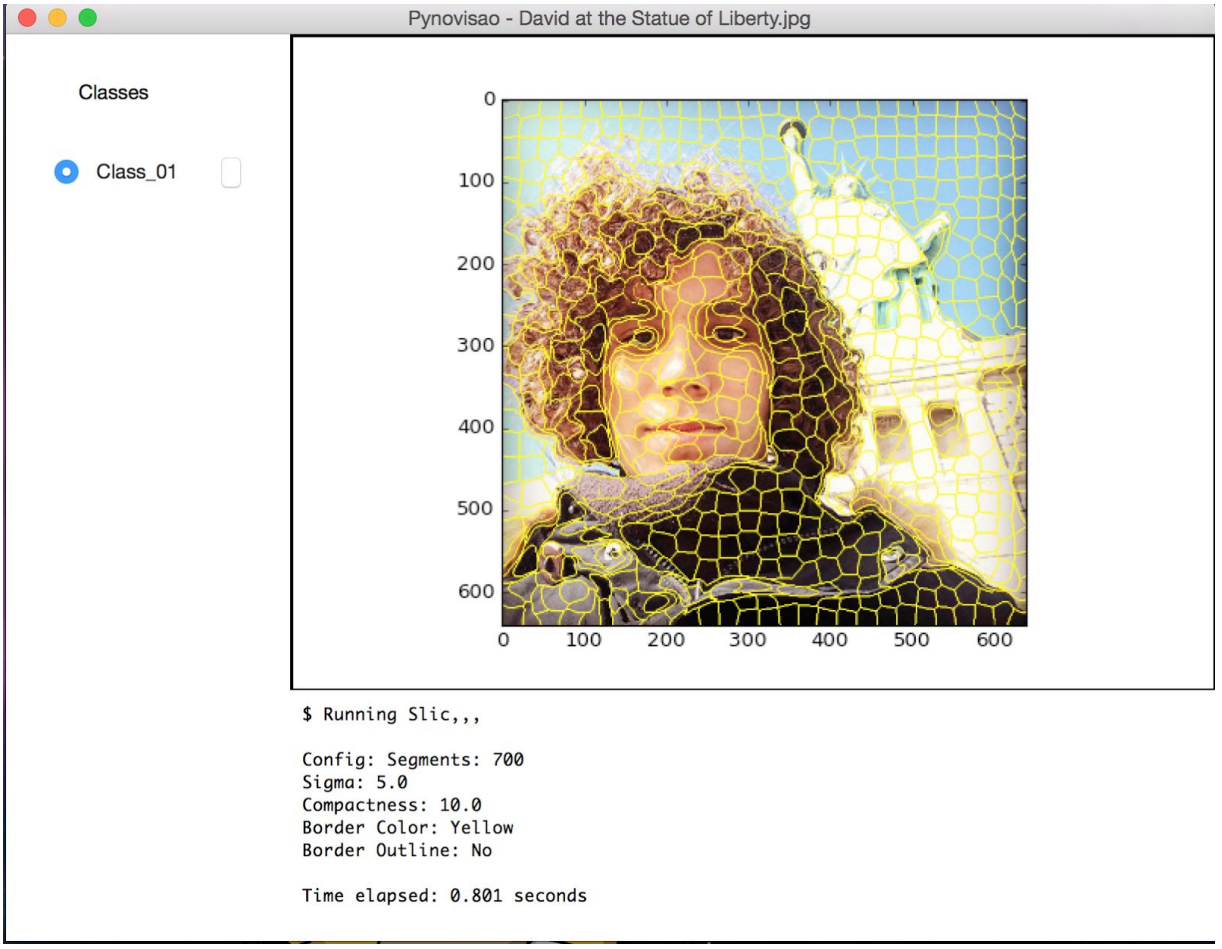

Figura 4: Ilustração de segmentação de imagens utilizando a técnica de *superpixels* através de uma implementação do algoritmo SLIC. Na imagem, o algoritmo é executado no *software* PYNOVISAO, desenvolvido pelo grupo INOVISAO.

No entanto, para que a técnica torne-se viável, o número de *superpixels* gerados deve ser bastante limitado e não apenas finito, uma vez que um número muito grande de *superpixels* em uma imagem maximizaria o custo computacional para a execução do algoritmo tornando-o ineficiente, lento e consequentemente inviável. Ainda, segmentar a imagem em um número demasiadamente grande de *superpixels* não faz sentido, uma vez que o número de *superpixels* se aproximaria do número de *pixels* da imagem. Sendo assim, um número usual de *superpixels* em uma imagem varia de uma média de 25 a 2500 *superpixels* (LINARES, 2013).

3.2.3. SLIC

Um dos métodos usados para extrair *superpixels* é chamado SLIC, usando um plano 5-D, com os parâmetros sendo os valores l,a,b do espaço de cores CIELAB, junto com as posições *x* e *y* de cada *pixel* para uma imagem de *N pixels*, contando *K superpixels*. O tamanho aproximado de cada *superpixel* é $\frac{N}{K}$, onde cada centro de *superpixel* está em um intervalo de $S=\sqrt{\frac{N}{K}}$ (SMITH, 2010). A função que determina a distância de cada *pixel* é dada pelas seguintes expressões:

$$D_{lab} = \sqrt{(l_k - l_i)^2 + (a_{k-}a_i)^2 + (b_k - b_i)^2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$D_s = d_{lab} + \frac{m}{S} * d_{xy}$$

As expressões acima constituem-se de funções feitas para a medição de distância entre os *superpixels* no plano 5-D. D_{lab} se constitui da distância dos *pixels* no plano de CIELAB, enquanto d_{xy} trata da distância euclidiana entre os *pixels* pela localização deles na imagem (SMITH, 2010).

Nas mesmas expressões, D_s é a função final que será usada como critério de distância entre os *pixels* no plano 5-D. Nesta função, pode-se determinar o quão compacto os *superpixels* são aumentando o valor de *m*, que dá mais ênfase à distância d_{xy}, normalizada pelo intervalo de grade *S*. O valor de m varia entre 1 e 20 (SMITH, 2010), pois valores maiores

que isso enfatizam à distância euclidiana entre os *pixels* na imagem, assim aglomerando apenas os *pixels* mais próximos uns dos outros, sem critério de cor e forma.

O método do SLIC foi escolhido pois já estava implementado no pynovisão e estava sendo usado por toda a equipe do INOVISAO.

3.2.4. Normalização dos atributos de superpixel antes de aplicar o k-means

A normalização dos atributos para o k-means é recomendada porque aplicar k-means com distâncias euclidianas em um banco de dados com dados muito dispersos pode afetar muito a localização dos centroides e arruinar os dados. É bem provável que este fator que estava fazendo todos os centroides muito distantes, assim fazendo com que todos os superpixels sejam uma classe.

Foi usado a normalização Z-score do Mohammad(2013), pelo framework Sklearn usando a função (PREPROCESSAMENTO).

A normalização viabilizou programa com histogramas que possuem informações mais claras e mais úteis para todos os métodos de machine learning que foram aplicados.

3.2.5. Normalização dos histogramas para a classificação das imagens

Foi usado o mesmo Z-score para fazer a normalização dos histogramas antes de passá-los para os classificadores. Isso tanto não afetou certos classificadores como viabilizou outros classificadores para a classificação de imagens, como será mostrado nos resultados.

3.3. Aprendizagem de Máquina

Na área da ciência da computação, a inteligência artificial é responsável pelo estudo de vários problemas de categorias diferentes. A inteligência artificial busca emular a capacidade humana de aquisição de informação, bem como seu processamento, reconhecimento de padrões e a capacidade do cérebro humano de aprender. Embora o conceito de inteligência em si e o que determina se um sistema computacional é de fato inteligente ou não seja alvo de discussões na academia, nos restringiremos apenas a seus exemplos, subáreas e o motivo de seus conceitos e técnicas serem relevantes para este trabalho.

Como dito anteriormente, a inteligência artificial trata de uma gama grande de problemas, como o reconhecimento de objetos através de padrões, filtros de *spam* em caixas de correios eletrônicos, jogos digitais, entre outros. Uma vez que trata-se de uma área presente em aplicações dos mais variados tipos, surgiu a necessidade de categorizar o campo da inteligência artificial. A aprendizagem de máquina ou aprendizagem automática (*machine learning*, em Inglês) é uma delas.

A aprendizagem de máquina é uma subárea da inteligência artificial que estuda e desenvolve algoritmos que sejam capazes de melhorarem seus respectivos desempenhos (medidos por sua porcentagem de acertos, por exemplo) perante um determinado problema, de acordo com sua própria experiência (SEWELL, 2007), análogo a um sistema de controle em malha fechada, ou seja, com realimentação. No caso da aprendizagem de máquina, os problemas de aprendizagem são categorizados. Estas categorias resumem-se em basicamente: aprendizagem supervisionada, não supervisionada, e semi-supervisionada.

3.3.1. Aprendizagem Supervisionada

No caso da aprendizagem supervisionada, são informados ao algoritmo dados de entrada e os referidos resultados esperados, que servem como exemplos. Este processo é conhecido como treinamento. Ou seja, ao fornecer exemplos de dados de entrada e as saídas esperadas ao algoritmo, dizemos que o mesmo está sendo treinado. Através dos exemplos apresentados manualmente ao algoritmo, o mesmo deve ser capaz então de generalizar os padrões apresentados nos exemplos através de uma função, que posteriormente será aplicada em exemplos não categorizados (dados em que os resultados esperados não são informados) (SEWELL, 2007).

Sendo assim, a partir do treinamento e da generalização feita através do processo de aprendizagem, o algoritmo então torna-se capaz de categorizar os resultados de exemplos de dados subsequentes passados como entrada. A função capaz de fazer a generalização dos dados a partir de exemplos é chamada de classificador. Do processo de treinamento, resulta-se um classificador. Dentre as técnicas e algoritmos de aprendizagem supervisionada, serão discutidos aqueles que são relevantes para este trabalho. São eles: SVM (SMO), Árvore de Decisão*, Random Forest,* MLP e Regressão Logística.

3.3.1.1. SVM (SMO)

Máquinas de Vetores de Suporte é uma técnica de aprendizagem de máquina supervisionada. A técnica foi desenvolvida por Vladimir Vapnik, que descreveu uma série de etapas que descrevem como conseguir bons classificadores. Como descrito anteriormente, do processo de treinamento obtemos uma função de generalização que implementa o classificador. Sendo assim, bons classificadores seriam aqueles que possuem uma boa função de generalização, afim de classificar corretamente novos dados (LORENA & CARVALHO, 2007).

Ainda, vale ressaltar que as SVMs baseiam-se na teoria de aprendizado estatístico. Lorena & Carvalho (2007) descrevem ainda que a técnica de SVM tem recebido destaque a uma atenção considerável da comunidade científica pelos bons resultados obtidos nas aplicações que utilizaram a técnica como base. A utilização da técnica de SVM justifica-se pelo fato de técnicas e algoritmos de classificação em aprendizagem de máquina serem utilizadas para fins importantes, como a classificação de um grande volume de dados (GIRARDELLO, 2010).

SMO é um algoritmo de SVM, ou seja, uma proposta de implementação prática dos conceitos contextualizados na descrição da técnica de SVM. É um algoritmo usado para resolver problemas de otimização, uma vez que baseia-se no conceito de divisão e conquista, bastante difundido em computação. Ou seja, um problema de complexidade computacional quadrática é dividido em uma série de outros menores e menos complexos, que serão posteriormente resolvidos de forma analítica (GIRARDELLO, 2010). Desta forma, podemos dizer que um problema com maior complexidade computacional é resolvido de forma mais simples, justificando assim a otimização.

3.3.1.2. Árvore de Decisão

Árvore de Decisão (*Decision Tree* em Inglês) é um dos algoritmos de árvores de decisão. Trata-se de uma estrutura utilizada para classificação de dados. A partir de uma coleção de dados de entrada a serem analisados (*data sets*), a estrutura da árvore é montada de acordo com os atributos utilizados para divisão de cada nó. A análise (e classificação) dos dados é realizada através do caminho que os referidos dados percorre na árvore, considerando

que a computação termina nas folhas da árvore, retornando, então, a decisão tomada de classificação para o referido dado de entrada.

Em outras palavras, as folhas da árvore de decisão referenciam as “classes” em que os dados podem ser classificados. Em cada nó não-folha é feita uma pergunta em relação ao dado de entrada. Cada resposta diferente corresponde a uma ramificação daquele nó, referenciando nós-filhos. A Figura 5 exemplifica o conceito através de uma árvore de decisão utilizada para concessão de bolsas de estudo. De acordo com os atributos expressos em cada nó da árvore, os alunos serão classificados em 2 categorias, expressas pelas folhas da árvore de decisão: aprovado ou reprovado no processo seletivo.

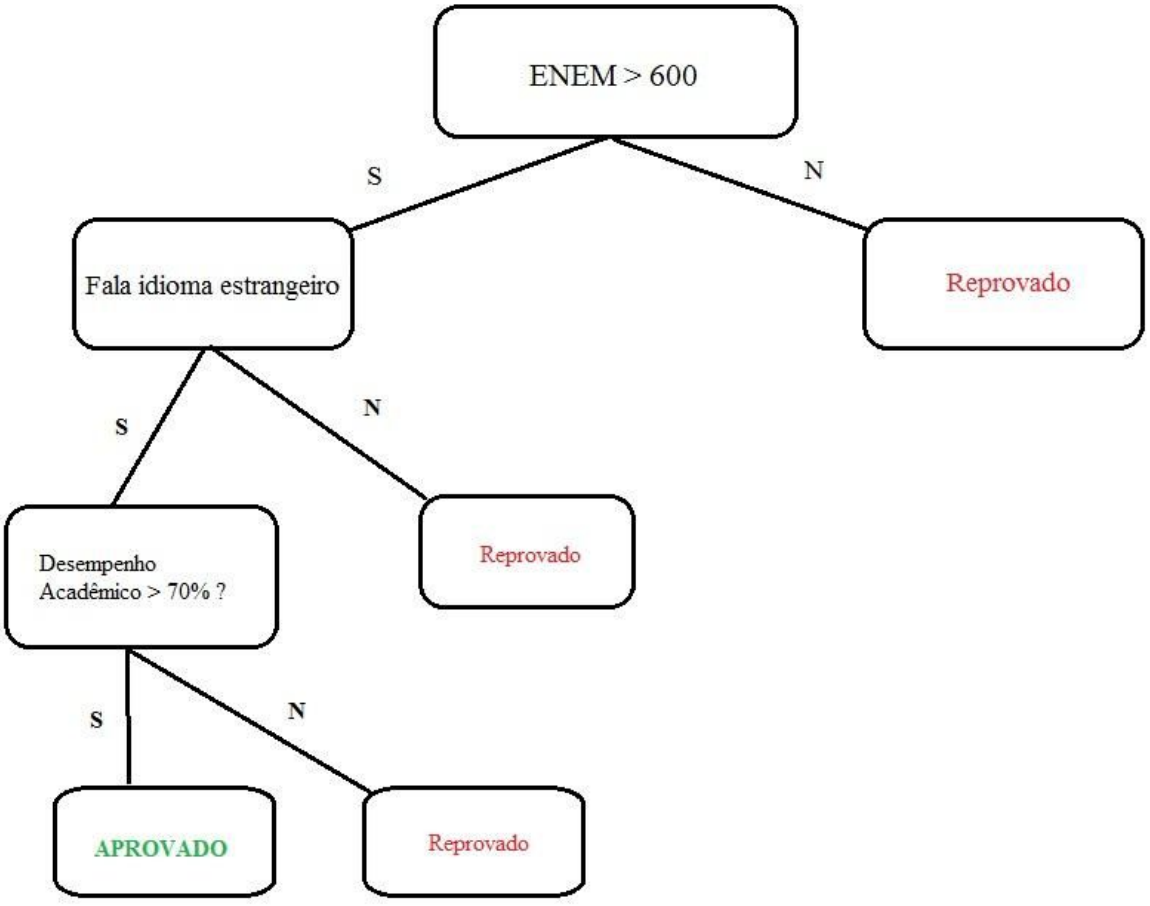

Figura 5: Árvore de decisão utilizada para concessão de bolsas de estudo.

Vamos entender o exemplo. Os dados de entrada começam a ser analisados a partir da raíz da árvore. A pergunta feita neste nó é: "o aluno possui nota no ENEM maior que 600?" Neste caso, temos apenas 2 respostas possíveis: sim ou não. Ambas respostas resultam em 1 nó-filho cada. Se a nota do aluno no ENEM for menor que 600, chegamos a uma das folhas na árvore. Assim sendo, o aluno é classificado e a computação é encerrada. Supondo que a resposta para o nó anterior (raiz) tenha sido "sim," seguimos com a análise do referido dado.

A pergunta do próximo nó é: "o aluno fala um idioma estrangeiro?" Novamente, temos 2 respostas possíveis: sim ou não. Novamente, ambas as respostas resultam em 1 nó-filho cada. Se a resposta for negativa para a pergunta, atingimos um nó-folha e a computação termina com a classificação do aluno. Se a resposta for positiva, avançamos mais um nó.

No próximo nó, a pergunta é: "o desempenho acadêmico do aluno é maior que 70%?" Em caso positivo, avançamos o nó da esquerda e cômputo resulta em uma classificação de aprovação para o aluno, e reprovação caso a resposta tenha sido negativa.

Como dito anteriormente e exemplificado através da Figura 5, cada folha da árvore corresponde a uma classe de classificação para os dados sendo analisados, e a quantidade de filhos de cada nó está diretamente relacionado ao atributo sendo considerado, ou "a pergunta sendo feita". Vale ressaltar que a estrutura da árvore de decisão é montada de acordo com dados pré-definidos, o que justifica ser um algoritmo de aprendizagem supervisionada.

Em outras palavras, a função que define o classificador neste caso é a estrutura da árvore em si, que é determinada levando em consideração o exemplos de treinamento. Alguns problemas a serem considerados em árvores de decisão são os atributos a serem considerados para as divisões dos nós, o que fazer caso a árvore tome um tamanho inviável para computação, entre outros.

3.3.1.3. *Random Forest*

Similarmente, *Random Forests* são junções de árvores de decisão. As árvores de decisão individuais são combinadas de forma a compor uma única *Random Forest*. Em outras

palavras, os resultados individuais de cada árvore de decisão são combinados de forma a compor o resultado final de uma *Random Forest*. A Figura 5 ilustra o conceito.

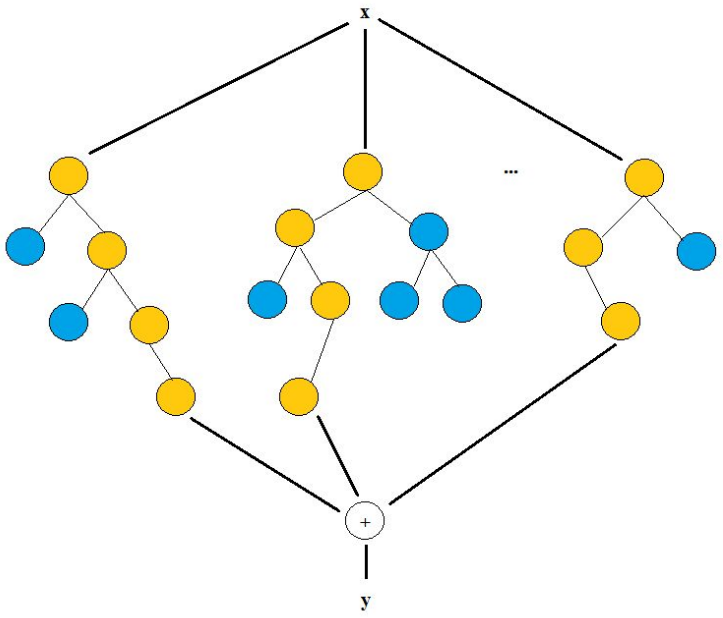

Figura 6: Ilustração de uma *Random Forest*.

3.3.1.4. MLP

MLP é uma técnica de aprendizagem de máquina que simula neurônios, tendo camadas de nós se comunicando entre si por ligações que modificam as saídas destes nós por pesos. O MLP funciona apenas com aprendizado supervisionado e precisa treinar com um conjunto de dados que já possui classes determinadas, como o nosso conjunto de dados de pólens.

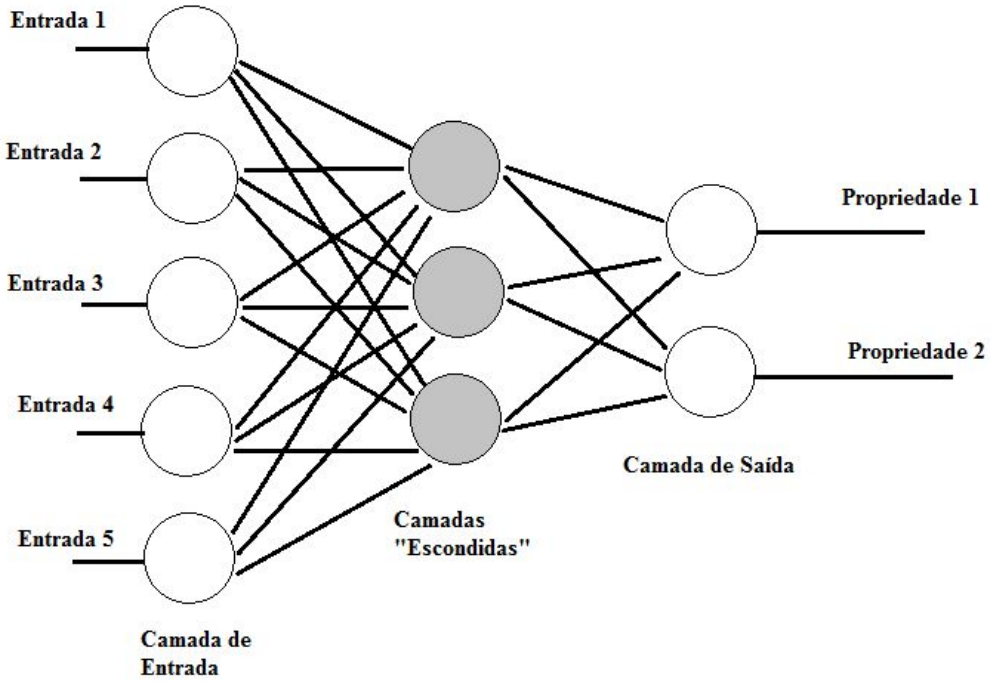

Figura 7: MLP.

Esta técnica começa construindo camadas de neurônios com pesos aleatórios e treina estas camadas automaticamente. O treinamento ocorre com o processamento direto e retro propagação. A processamento direto consiste apenas em inserir os dados de entrada do banco de dados e ver o resultado atual A. Com A, calcula-se o erro E em relação à resposta correta R.

Com E, proporcionalmente a taxa de aprendizado T, é realizada a retro propagação que calcula os novos pesos para a rede de neurônios. O objetivo do programa é minimizar E. Este processo se repete em épocas, nome para processamento direto e retro propagação juntos.

É importante ressaltar que há um custo tanto para uma taxa de aprendizagem pequena quanto para uma grande. Se esta taxa for muito grande, o neurônio acaba se afastando do mínimo global do erro e nunca chegando no resultado desejado, enquanto se esta taxa for pequena, o tempo gasto para chegar no mínimo do erro é muito alto (RUMELHART, 1986).

3.3.1.5. Regressão Logística

A regressão logística é um ótimo classificador para determinar problemas classificativos como a probabilidade de uma pessoa ter câncer, por exemplo. Este exemplo contém apenas uma classe como resposta, ter câncer.

A regressão logística funciona usando a função sigmoide para classificar os parâmetros e usa uma função de custo inversa a esta sigmoide para modelar o classificador como mostrado na Figura 8.

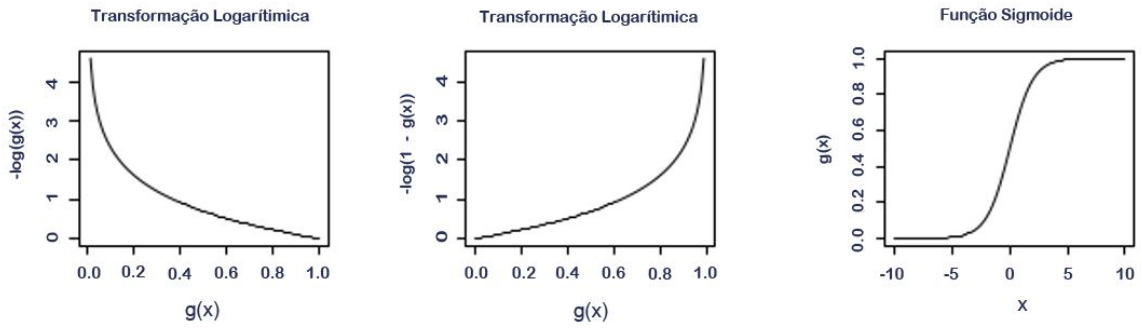

Figura 8: Sigmoide e funções de custo.

Na Figura 8, o gráfico (c) (último da esquerda para direita) seria a sigmoide que classifica um exemplo, ela funciona em função de todos os parâmetros deste exemplo e retorna *g(x)*, que é a possibilidade do exemplo ter câncer.

Ainda na Figura 8, os gráficos (a) e (b) são as funções de custo para quando o exemplo testado não ter e ter câncer. Estes gráficos estão em função de *g(x)*.

Observando o gráfico (b), representando a situação em que o exemplo não tem câncer, percebe-se que quanto maior o *g(x)*, maior o custo com o gráfico tendendo ao infinito. Usando as funções de custo, o programa usa várias iterações para modelar o classificador e minimizar o erro baseado em todos os exemplos dados para ele.

Para o problema abordado neste trabalho, ou seja, um problema de múltiplas classes, faz-se um modelo de regressão linear para todas as classes. A classificação é feita usando todos os modelos e escolhendo a classe que teve maior *g(x)* (HOSMER, 2000). O gráfico (b) tem custo de 0 para *y*, enquanto o gráfico (a) tem custo 1 para *y*.

3.3.2. Aprendizagem Não Supervisionada

Diferentemente da aprendizagem supervisionada onde as informações das classes do problema são conhecidas, a aprendizagem não supervisionada trata do oposto, ou seja, problemas onde as informações das classes são desconhecidas (SILVA, 2017). Isso consiste em dizer que, neste caso, os dados de entrada são informados ao algoritmo mas os resultados esperados são desconhecidos ou não informados. Em outras palavras, um modelo ou função de generalização que implementa o classificador é criado diante das "observações" aos dados de entrada do problema, somente (SEWELL, 2007). Um clássico exemplo de aprendizagem não supervisionada são os algoritmos de aglomeração de dados (*clustering*). Neste trabalho, o algoritmo K-médias é utilizado para categorizar os *superpixels* gerados a partir do processo de segmentação da imagem em classes.

3.3.2.1. Algoritmo K-médias

Em vários campos de pesquisa, como taxonomistas, cientistas sociais, psicólogos, biólogos, estatísticos, matemáticos, engenheiros e pesquisadores médicos, há muito uso de aglomeração de dados para classificação dos mesmos (JAIN, 2010). Neste trabalho, é usado aglomeração de dados em visão computacional com o K-médias.

K-médias é um algoritmo tradicional de inteligência artificial onde uma de suas limitações é expressa pelo fato de termos de escolher o número de centroides que temos que usar. Um dos objetivos de qualquer pesquisa em K-médias é saber o número correto *k* de centroides a serem usados para classificar as imagens corretamente. Os centroides são pontos em um espaço de *d* dimensões, onde os dados de nossa escolha residem.

A função abaixo representa uma função local, aplicada a cada um dos *k* centroides para a classificação de cada elemento baseado no erro quadrático entre cada elemento e um centroide. O objetivo é escolher os elementos mais pertos para o *k* centroide e minimizar o $J(c_k)$ (JAIN, 2010).

$$J(c_k) = \sum_{xi \in Ck} (x_i - \mu_k)^2$$

Sendo $J(c_k)$ o custo de cada centroide *k*, x_i a posição de cada elemento dos dados de treinamento e μ_k, a posição de cada centroide *k*.

O objetivo do K-médias é achar o mínimo global dos erros quadráticos entre os centroides e todos os elementos em cada conjunto, este representado pela equação abaixo (JAIN, 2010).

$$J(c) = \sum_{k-1}^{k} \sum_{xi \in Ck} (x_i - \mu_k)^2$$

Sendo J(c) a soma dos custos de todos os centroides feitos pelo K-médias.

No entanto, o K-médias apenas acha mínimos locais na função *J(c)*, por isso precisa-se fazer vários testes com diferentes *k* para se definir o melhor *k* para esse tipo de experimento.

K-médias é usado apenas para agrupar cada *superpixel* do banco de imagens em *k* grupos. Com os *superpixels* categorizados, cada imagem terá um histograma com a quantidade de tipos de *superpixels* que cada imagem contém.

3.3.2.2. *Mini Batch K-means*

A técnica de *Mini Batch* implementa uma versão do algoritmo K-médias para uma quantidade massiva de dados. Neste caso, é definido um "teto" de dados a serem analisados por cada iteração, de forma que o custo computacional do programa é reduzido, juntamente com a quantidade de dados analisados por iteração. Em outras palavras, nem todos os dados do conjunto de dados é analisado, mas sim, um número fixo de amostras.

Como o número de amostras (ou dados) é fixo e reduzido, os cálculos de distância entre as amostras também é reduzido, justificando, assim, a redução de custo computacional. No entanto, uma vez que o número de amostras é reduzido afim de reduzir o custo computacional do algoritmo, a "qualidade" da aglomeração dos dados também é reduzida (BÉJAR, 2013).

Foi usado o *Mini Batch K-means* por causa dos requerimentos computacionais do K-médias. Usando o K-médias de dois *frameworks* diferentes, o *framework* WEKA e o *framework Scikit-Learn*, o *Scikit* estava utilizando muito espaço de memória RAM e o WEKA se mostrou muito ineficiente porque ele não tem suporte de processamento multi núcleo.

O motivo pelo qual o *Batch K-means* ou K-médias convencional demorou para processar foi o fato de sua complexidade ser O(kns), onde *k* é o número de agrupamentos, *n* é o número de exemplos e *s* é o número máximo de elementos não zero (SCULLEY, 2010).

Tanto Sculley (2010) quanto Feizollah (2014) fizeram testes comparando o K-médias convencional e o *Mini Batch K-means*, e os dois concluíram que o *Mini Batch K-means* é mais eficiente e não há uma diferença significante na soma das distâncias entre os centroides e os dados.

3.3.3. Aprendizagem Semi-Supervisionada

Como visto até o momento, na aprendizagem supervisionada as entradas e resultados esperados do problema são conhecidos, enquanto na aprendizagem não supervisionada, acontece o oposto: os resultados para as dadas entradas não são conhecidos. A aprendizagem semi-supervisionada consiste em trabalhar com informações parcialmente conhecidas das classes, assim como seu próprio nome sugere. Seria uma classe intermediária de aprendizagem automática que se encontra entre a supervisionada e a não supervisionada, baseando-se em características de ambas (CHAPELLE, 2006).

No caso do histograma de superpixels, a parte não supervisionada é a separação dos superpixels em classes pelo K-means, logo a montagem do histograma em si é executada automaticamente. No entanto, as classes do banco de imagens de treinamento são conhecidas e classificadores de aprendizagem supervisionada são treinados com os histogramas de superpixels e suas classes.

Logo, uma aprendizagem semi-supervisionada pode criar novos parâmetros para ajudar na classificação de amostras com as classes conhecidas.

3.4. Métricas de Classificação

3.4.1. PCC

Esta métrica determina a taxa de acertos de classificação em relação a uma coleção de imagens (SILVA, 2017). Em outras palavras, esta métrica estabelece uma relação entre as imagens classificadas de maneira correta em relação ao número total de imagens. No entanto, é importante destacar que as imagens são classificadas em grupos, e a relação matemática que estabelece a relação comentada anteriormente é a seguinte:

$$PCC = \frac{VP+FP}{VP+FP+VN+FN}$$

Sendo que VP corresponde ao grupo dos verdadeiros positivos, FP corresponde ao grupo dos falsos positivos, VN corresponde ao grupo dos verdadeiros negativos e FN corresponde ao grupo dos falsos negativos (SILVA, 2017). Uma forma simplificada de representar esta equação é a seguinte:

$$PCC = \frac{Total\ classificado\ corretamente}{Total\ de\ imagens}$$

3.4.2. Medida-F

A medida-F expressa uma razão entre precisão e revocação (ou sensibilidade) (SILVA, 2017), e o seu fim de utilização é a comparação da função de generalização de classificadores (GONÇALVES, 2015). A referida razão entre precisão e revocação é expressa pela equação abaixo:

$$F = 2 * \frac{Precisão * Revocação}{Precisão + Revocação}$$

Sendo que a precisão, que é calculada para cada classe (SILVA, 2017), é expressa pela equação a seguir:

$$p = \frac{VP}{VP+FP}$$

A equação da precisão expressa o conceito de que a precisão é a razão entre aqueles que são classificados como pertencentes a uma referida classe e aqueles que são previstos como sendo daquela classe (SILVA, 2017). Por outro lado, revocação ou sensibilidade é determinada pela razão entre aqueles que foram corretamente classificados como positivos e o total de amostras que de fato são positivas (SILVA, 2017). Este conceito é expresso pela equação abaixo:

$$r = \frac{VP}{VP+FN}$$

3.4.3. Matrizes de Confusão

A matriz de confusão é uma representação gráfica de uma relação de no mínimo 2 atributos com suas respectivas frequências. Ainda, de acordo com Gonçalves (2015), as cores das matriz de confusão são estabelecidas de acordo com o método termal de Lee (2005), que relaciona as cores com a energia térmica remanescente de corpos ou objetos através de

transferência de calor por radiação, a qual está diretamente relacionada com a energia cinética molecular média do corpo ou objeto em questão. A energia que é irradiada dos corpos é capturada através de infravermelho. No caso, os tons mais escuros representam o maiores valores. Por exemplo, tons de vermelho representariam os maiores valores (de comprimento de onda), valores médios seriam alaranjados, e por fim, menores amplitudes seriam expressas por amarelo (GONÇALVES, 2015).

3.5. Técnicas de Amostragem

3.5.1. Validação Cruzada

A validação cruzada é uma técnica de amostragem que tem por objetivo avaliar quão boa é a função de generalização de um classificador. Em outras palavras, é uma técnica que avalia a capacidade de predição de um classificador.

Isto é feito a partir da divisão dos dados em dois grupos: os dados a serem utilizados no treinamento, e os dados a serem utilizados nos testes. Supondo um grupo *n* de dados do conjunto de dados, um subconjunto de *n*-1 dados são utilizados para treinamento, enquanto um grupo menor é utilizado para testar o classificador (SANTOS et al., 2009). Por exemplo, considerando que o conjunto de dados de entrada a serem analisados seja $n = 4$, $n-1=3$ conjuntos é utilizado no treinamento enquanto 1 subgrupo é utilizado para testes. Vale ressaltar que o conjunto de dados para treinamento deve ser maior do que o conjunto de dados para teste.

O processo todo acontece *n* vezes, sendo que a cada iteração, um conjunto específico de dados é o utilizado para testes. Ao final, é feita uma média de todos os resultados obtidos a partir de cada iteração para chegar a uma estimativa do classificador (SANTOS et al., 2009).

4. HISTOGRAMA DE *SUPERPIXELS*

O objetivo final deste trabalho é montar uma estrutura que utiliza histograma de *superpixels*. O programa é constituído de basicamente duas partes: a primeira, extração de atributos de cada um dos *superpixels* da imagem segmentada, e a segunda, construção do histograma em si.

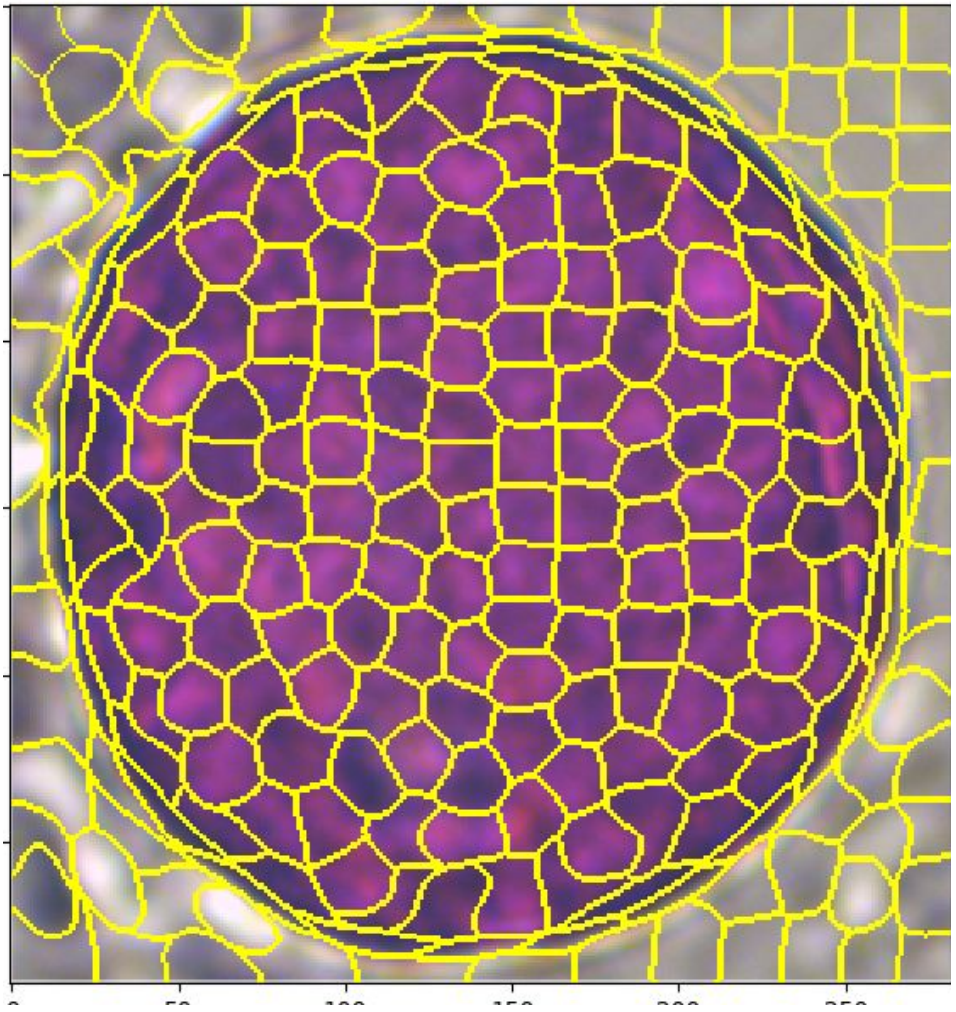

Figura 17 – Pólen segmentado em superpixels

Sendo assim, faz-se necessário primeiramente segmentar a imagem que desejamos analisar. Cada imagem é constituída de um pólen. A imagem é segmentada através da técnica de *superpixels* utilizando o algoritmo SLIC, assim como demonstrado na Figura 17. Após a imagem ser segmentada, são extraídos atributos de cada um de seus segmentos, ou *superpixels*, neste caso. A extração de atributos de cada um dos *superpixels* é feita através de extratores de atributos sumarizados de cor, matriz de co-ocorrência de níveis de cinza,

histograma de gradientes orientados, e padrões binários locais, que são os padrões do PYNOVISAO.

Uma vez que os atributos foram extraídos dos *superpixels*, estes conjuntos de atributos ou informações são postas em um plano de n dimensões, sendo n os atributos do superpixel, afim de aplicar o algoritmo K-médias neste plano, encontrando, assim, um número *k* de centroides, que serão os pontos de referência para a classificação de cada classe de *superpixel*. Após isso, cada *superpixel* é então classificado de forma individual.

Para cada imagem, então, é construído um histograma de *superpixels*. Com todos os histogramas correspondentes a cada imagem do banco de dados, o classificador com aprendizado de máquina supervisionado é treinado para classificar novos histogramas de *superpixel*.

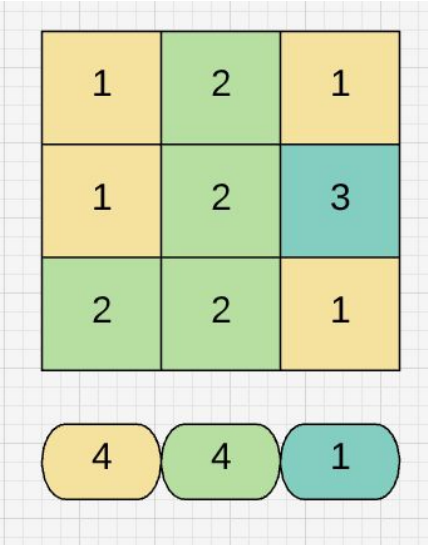

Figura 18: Exemplo de uma imagem com o seu histograma de *superpixel*

Digamos que a Figura 18 é uma imagem de pólen com 9 superpixels e configuramos o k do histograma para 3, pode-se observar que o k-médias achou 4 superpixels da classe 1, 4 da classe 2 e 1 da classe 3, com estas informações ele organizou o histograma de superpixels.

Com o classificador pronto, basta passar novas imagens de pólens para o mesmo. O classificador então realiza a classificação de um único pólen na imagem pela quantidade de *superpixels* de cada tipo presente na imagem.

A seguir, o algoritmo proposto é apresentado em pseudocódigo (também chamado de português estruturado ou portugol), afim de melhor demonstrar o processo descrito em texto.

```
algoritmo "extrai_atributos_superpixels"
var
        //declaração de variáveis
        n: inteiro;
        m: inteiro;
        imagem_original: arquivo_de_sistema;       //entrada do algoritmo
        imagem_segmentada: arquivo_de_sistema;
        arquivo_de_atributos: arquivo_de_sistema; //saída do algoritmo
inicio
        n <- numero_de_imagens;
        m <- numero_de_superpixels;
        para i <- 0 ate n - 1 passo 1 faca
                imagem_segmentada <- segmenta_imagem_slic(imagem_original);
                para j <- 0 ate m-1 passo 1 faca
                        arquivo_de_atributos <- extrai_atributos(imagem_segmentada[m]);
                fimpara
        fimpara
fimalgoritmo
```

Figura 9: Algoritmo para extrair atributos de cada *superpixel* da imagem.

.

```
algoritmo "histograma_superpixels"
var
        //declaração de variáveis
        n: inteiro;
        m: inteiro;
        k: inteiro;
        x: inteiro;
        vetor_de_classes[x]: vetor;
        arquivo_de_atributos: arquivo_de_sistema;        //entrada do algoritmo
        arquivo_de_histograma: arquivo_de_sistema;       //saída do algoritmo
```

Figura 10: Primeira parte do algoritmo para achar os histogramas por imagem na pasta.

```
inicio
        n <- numero_de_imagens;
        m <- numero_de_superpixels;
        para i <- 0 ate n - 1 passo 1 faca
                //cria-se um vetor de zeros para cada imagem
                //cada posicao do vetor representa uma classe;
                para j <- 0 ate m - 1 passo 1 faca
                        k <- kmeans(arquivo_de_atributos); //numero k de centroides
                        //determina-se a classe 'x' do superpixel;
                        //incrementa a posição da referida classe no histograma;
                        vetor_de_classes[x] <- vetor_de_classes[x] + 1;
                fimpara
                arquivo_de_histograma <- cria_histograma(vetor_de_classes);
        fimpara
fimalgoritmo
```

Figura 11: Segunda parte do algoritmo para achar os histogramas por imagem na pasta.

Agora com os histogramas, pode-se usar técnicas de aprendizado de máquina para classificar novas imagens contendo um pólen.

Pensando em uma aplicacao real desta tecnica, voce teria um fluxo de trabalho como mostrado na Figura 19.

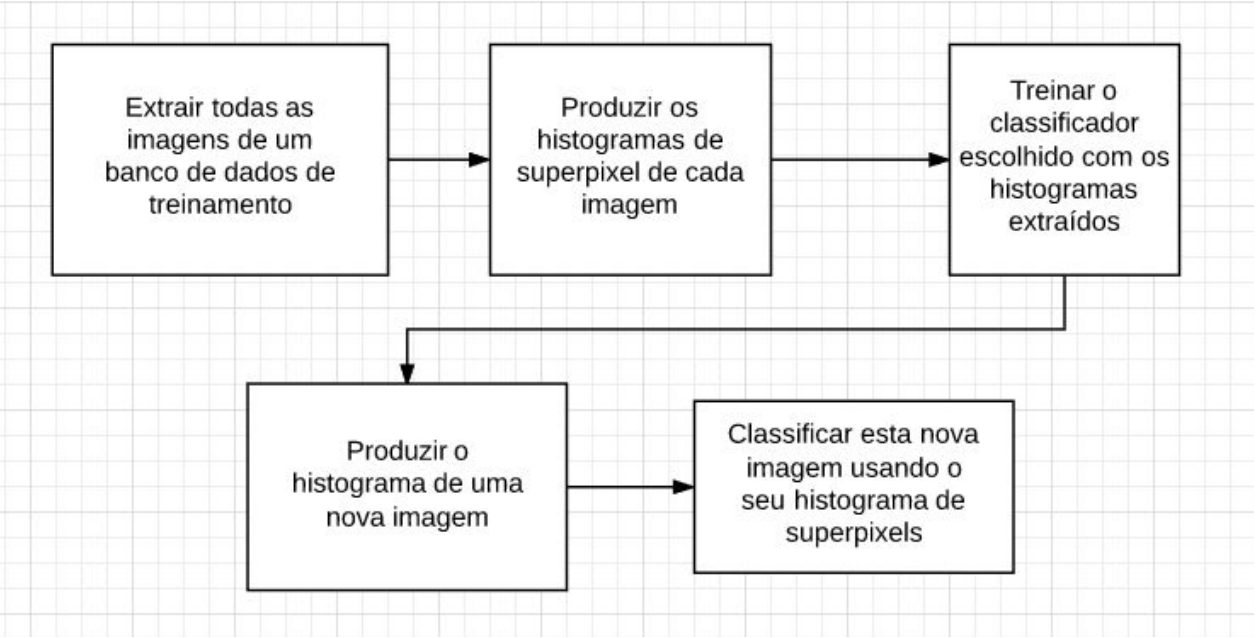

Figura 19: Fluxo de trabalho do histograma de superpixels em uma aplicação real

5. METODOLOGIA

Para cada um dos objetivos específicos listados no capítulo 2, são apresentados a seguir os aspectos metodológicos que nortearam a execução deste trabalho.

5.1. Estudo Detalhado da Técnica de Segmentação por *Superpixels* Baseada no Algoritmo SLIC

Como metodologia de revisão de literatura para as técnicas e algoritmos citados e estudados durante o desenvolvimento deste trabalho, foi adotada a revisão narrativa. Uma vez que a revisão narrativa não utiliza de critérios explícitos e sistemáticos para a busca de literatura, um dos primeiros passos para encontrar bons materiais relacionados aos temas deste trabalho foi o próprio acervo do grupo INOVISAO.

No grupo foram desenvolvidos trabalhos correlatos cuja leitura nos guiou sobre a melhor compreensão do problema a ser resolvido, bem como referenciou outros materiais também relacionados aos assuntos aqui discutidos, como por exemplo palinologia, visão computacional, segmentadores de imagens, e inteligência artificial.

Além disto, consultas aos principais portais de periódicos mundiais, como *IEEE Xplore*, ACM DL, *Science Direct* e *Scopus*, ajudaram a identificar artigos com trabalhos correlatos nas áreas de segmentação por *superpixels* e palinologia forense. Estes artigos foram revisados para complementar o entendimento da proposta aqui discutida.

5.2. Implementação do Novo Extrator de Atributos "Histograma de *Superpixels*"

O módulo foi desenvolvido em linguagem Python, que foi escolhida por apresentar uma sintaxe de fácil aprendizado, bem como considerável desempenho computacional. Como citado anteriormente, um *software* na mesma linguagem já foi desenvolvido pelo grupo INOVISAO anteriormente: o PYNOVISAO.

Dentre as técnicas implementadas previamente no referido *software*, a implementação referente ao segmentador SLIC foi reaproveitada no trabalho proposto, utilizando os códigos

do módulo deste segmentador no PYNOVISAO. Ainda, foram seguidas as regras definidas pelo grupo de pesquisa e desenvolvimento INOVISAO disponíveis no site do grupo[2].

5.3. Validação do Módulo Implementado

Após a implementação do algoritmo "Histograma de *Superpixels*," proposto neste trabalho, os resultados obtidos foram comparados aos algoritmos já implementados no *software* anteriormente desenvolvido pelo grupo INOVISAO, o PYNOVISAO, tendo como base o banco de imagens de grãos de pólen também remanescente de trabalhos prévios correlatos.

Para cada algoritmo testado, foram calculados os desempenhos médios, e então foi constatado o quanto os resultados dos algoritmos diferem estatisticamente, comprovando as vantagens de uns em relação aos outros. Desta forma, foi determinada a Taxa de Classificação Correta (*Correct Classification Rate - CCR*) do *software*, que é uma das métricas de desempenho aplicadas. Ainda, outras métricas de desempenho utilizadas foi Medida-F (*F-Measure*) e Matrizes de Confusão. Como técnica de amostragem foi utilizada Validação Cruzada, que é uma das técnicas comumente aplicadas em projetos desenvolvidos no grupo INOVISAO, e que inclusive já foi aplicada em trabalhos correlatos.

Vale ressaltar que o PCC (ou TCC – Taxa de Classificação Correta, como foi chamado anteriormente) e medida-F são comumente aplicados a problemas de 2 classes, como sugere a revisão de literatura. No entanto, o problema abordado nesse trabalho lida com mais de 2 classes. Quanto a adaptação de ambos para problemas envolvendo mais de 2 classes podemos dizer que ambos são calculados em base da taxa de acerto de cada classe individualmente, da mesma forma que seria feito em problemas de 2 classes.

5.4. Integração com Sistema Já Existente (PYNOVISAO)

A integração é feita por meio da modificação do módulo já implementado anteriormente. Exemplos de modificações a serem citados é a adição de recursos na interface gráfica para que as técnicas implementadas fiquem disponíveis para que o usuário do sistema possa fazer uso das mesmas. Os códigos referentes as implementações das novas técnicas

[2]O site do INOVISAO está em www.gpec.ucdb.br/inovisao

foram enviados ao repositório do grupo INOVISAO no Git, no diretório do módulo anteriormente implementado.

Após isto, foram efetuados testes para a verificação de eventuais erros na integração, como por exemplo a utilização do *software* quanto a todas suas funcionalidades, antigas e novas. Uma vez que constatados comportamentos anômalos quanto a quaisquer das técnicas disponíveis no sistema, eventuais erros de integração foram verificados diretamente no código da referida técnica. Por fim, após a correção de eventuais erros, a integração ao sistema foi concluída.

6. RESULTADOS E DISCUSSÃO

O banco de dados de imagens é formado por 68 classes de pólens com 35 imagens cada. Cada imagem foi dividida em 250 *superpixels*. De cada *superpixel* foram extraídos 218 atributos. O banco de dados foi dividido 70/30 (para validação cruzada) entre os dados de treinamento e os dados de teste.

Abaixo estão apresentados as medidas-F dos dados de treinamento que sofreram validação cruzada com 5 dobras. Esta validação cruzada não leva em conta os atributos K-médias, ela só é aplicada nos histogramas já prontos. É uma validação cruzada apenas para os classificadores. Este fator é importante porque isso quer dizer que os histogramas foram formados com um K-médias construído com dados que ele não deveria ter.

O objetivo desta validação cruzada de dados de treinamento era escolher o melhor *k* (número de centroides para o K-médias) e o melhor classificador de dados.

Tabela 1. Medidas-F de k no intervalo entre 10 e 100 aplicando validação cruzada nos dados de treinamento normalizados.

k	**Random Forest**	**Logistic Regression**	**MLP**	**Decision Tree**	**SVM**
10	0,53	0,51	0,36	0,46	**0,56**
25	0,63	**0,67**	0,56	0,47	**0,67**
40	0,66	0,73	0,64	0,53	**0,74**
55	0,61	0,65	0,58	0,53	**0,74**
70	0,67	**0,75**	0,62	0,54	0,71
85	0,66	0,75	0,68	0,55	**0,79**

100	0,67	**0,78**	0,7	0,5	0,66

Tabela 2. Medidas-F de k no intervalo entre 10 e 100 aplicando validação cruzada nos dados de treinamento não-normalizados.

k	Random Forest	Logistic Regression	MLP	Decision Tree	SVM
10	**0,56**	0,54	0,46	0,44	0,17
25	0,62	**0,64**	0,5	0,49	0,24
40	0,63	**0,65**	0,6	0,5	0,33
55	0,65	**0,68**	0,6	0,54	0,37
70	0,66	**0,75**	0,66	0,54	0,45
85	0,68	**0,75**	0,67	0,57	0,49
100	0,66	**0,77**	0,66	0,58	0,56

Pode-se observar que a normalização dos histogramas não afetou muito nenhum método de aprendizado, porém ajudou muito o SVM a classificar os *superpixels*. Foi escolhido o SVM como melhor classificador de *superpixels* e k=85 foi escolhido como o melhor *k* em um intervalo de 10 a 100. É importante destacar que valores maiores do que 100 se mostraram irrelevantes.

Tabela 3. Medidas-F dos resultados do SVM com k=85 aplicado nos dados de teste normalizados.

Random Forest	0,7099
Logistic Regression	0,7848
MLP	0,7754
Decision Tree	0,5802
SVM	0,8302

A decisão de usar o SVM com k=85 foi inteiramente baseado nos testes com a validação cruzada. A tabela 3 mostra os resultados do classificador treinado nos dados de treinamento classificando os dados de teste. Com a classificação dos dados de teste, pode-se tirar a conclusão que a validação cruzada não foi adulterada por causa das informações que o K-médias não deveria ter.

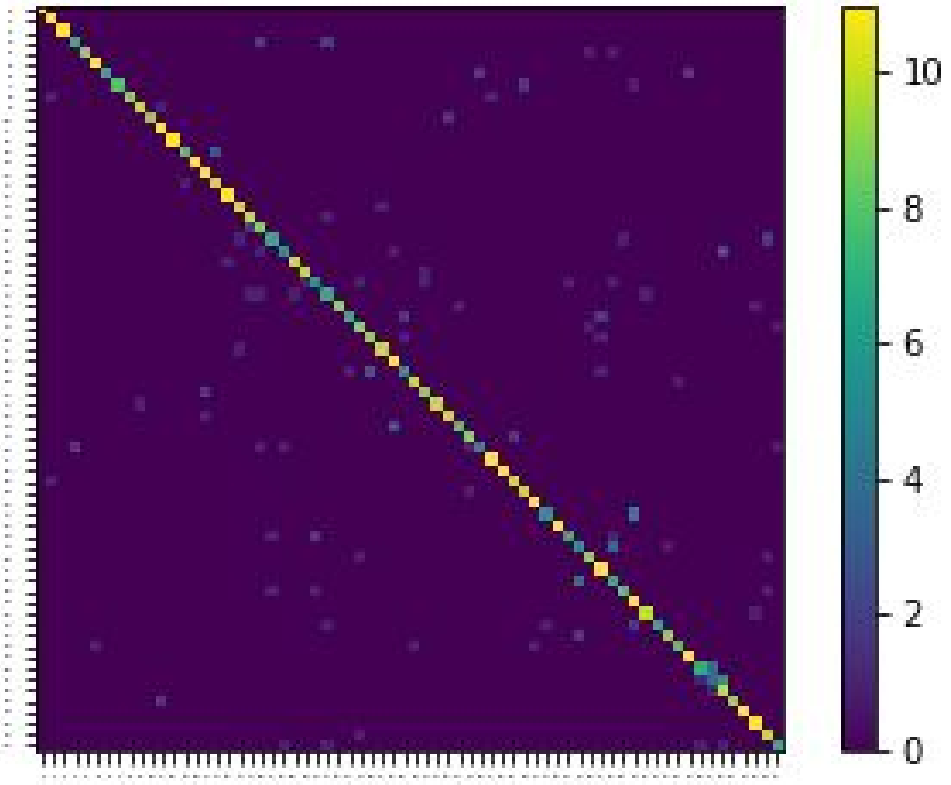

Figura 12: Matriz de confusão do SVM com k=85 aplicado nos dados de teste.

Tabela 4. PCC de todas as classes de pólens testadas.

Polen	SVM	MLP	Decision Tree	Logistic Regression	Random Forest
acoita_cavalo	1.00	0.91	1.00	1.00	1.00
acrocomia_aculeta	0.90	0.36	0.55	0.64	0.82
anadenanthera_colubrina	1.00	0.82	0.82	1.00	1.00
arachis	0.91	0.36	0.45	0.45	0.73
arrabidaea_florida	0.82	0.82	0.73	0.91	0.82
aspilia_gracielae	1.00	0.73	0.82	1.00	0.82
bacopa_australis	0.91	0.64	0.64	1.00	0.64
beladona	0.82	0.55	0.36	0.73	0.82
caesalpinia_peltophoroides	0.64	0.73	0.64	0.73	0.64
caryocar_brasiliensis	0.91	0.82	0.64	0.91	0.73
cecropia_pachystachya	1.00	0.82	0.82	1.00	0.73
ceiba_speciosa	1.00	0.91	0.55	1.00	0.91
chromolaena_laevigata	1.00	1.00	0.64	0.91	0.91
cissus_campestris	0.73	0.73	0.55	0.82	0.73
cissus_spinoza	1.00	0.73	0.73	1.00	0.91
combretum_discolou	1.00	0.91	0.82	1.00	0.82
cordia_trichotoma	0.91	0.82	0.82	0.55	0.91
cosmos_caudatus	1.00	0.73	0.91	1.00	1.00
croton_urucurana	0.82	0.73	0.73	0.91	0.91
curva_lago	0.91	0.73	0.36	0.91	0.91
dianella	0.91	0.45	0.55	0.64	0.36
dipteryx_alata	0.82	0.45	0.09	0.36	0.27
doliocarpus_dentatus	0.73	0.45	0.18	0.27	0.36
erythina_dominguesi	1.00	0.73	0.82	0.91	0.73
eucalyptus_sp	0.91	0.73	0.73	0.91	0.73
faramea_sp	0.73	0.09	0.55	0.45	0.36
genipa_auniricana	0.64	0.45	0.18	0.73	0.64
guazuma_ulmifolia	0.73	0.73	0.82	0.82	0.82
hortica_areadica	0.64	0.55	0.64	0.55	0.73
hyptis_sp	0.91	0.82	0.55	0.82	0.18
ligustrum_lucidum	0.91	0.55	0.55	0.91	0.36
mabea_fistulifera	0.91	0.82	0.64	0.82	0.73
machaerium_aculeatum	0.91	0.91	0.82	1.00	0.73
magnolia_champaca	0.82	0.27	0.64	0.91	0.27
mandioca	1.00	0.64	1.00	0.91	1.00
matayba_guianensis	0.55	0.27	0.27	0.45	0.18
mauritia_flexuosa	0.91	0.73	0.82	1.00	0.91
mimosa_ditans	1.00	0.64	0.55	0.64	0.73
mimosa_pigra	1.00	0.36	0.36	0.82	0.64
mitostemma_brevifilis	0.82	0.45	0.55	1.00	0.91
myracroduon_urundeuva	0.36	0.45	0.36	0.27	0.27
ochroma	1.00	0.91	0.73	1.00	1.00
ouratea_exasperma	1.00	1.00	0.82	1.00	0.91
pachia_aquatica	0.91	1.00	0.64	0.73	0.73
palmeira_real	0.91	0.45	0.55	0.73	0.82
passiflora_giberti	0.73	0.55	0.55	0.82	0.36
paullinia_spicata	0.82	0.27	0.82	0.45	0.82
piper_aduncum	1.00	1.00	1.00	1.00	1.00

poaceae_sp	0.64	0.45	0.36	0.64	0.55
protium_heptaphyllum	0.73	0.27	0.45	0.45	0.36
qualea_multiflora	0.73	0.55	0.27	0.73	0.45
ricinus	1.00	0.64	0.82	1.00	0.91
schinus_sp	0.45	0.45	0.45	0.45	0.55
senegalia_plumosa	0.73	0.73	0.18	0.82	0.36
serjania_erecta	1.00	0.82	0.55	0.82	0.82
serjania_hebecarpa	1.00	0.91	0.36	0.91	0.82
serjania_laruotteana	0.64	0.73	0.18	0.64	0.09
serjania_sp	0.91	0.91	0.55	0.91	0.82
sida_cerradoensis	0.64	1.00	0.73	0.64	0.73
solanum_sisymbrifolium	1.00	0.91	0.73	0.91	0.82
syagrus_romanzoffiana	0.73	0.36	0.55	0.36	0.55
tabebuia_chysotricha	0.82	0.36	0.36	0.27	0.64
tabebuia_rosealba	0.73	0.36	0.55	0.91	0.45
tradescantia_Pallida	0.91	0.64	0.73	0.64	0.82
trema_micrantha	1.00	0.64	0.91	1.00	1.00
trembleya_phlogiformis	1.00	0.82	1.00	1.00	0.82
tridax_procumbens	0.82	0.73	0.64	0.91	0.45
vochysia_divergens	0.55	0.27	0.18	0.45	0.27

Baseado no PCC de cada classe, 3 classes foram selecionadas para serem estudadas mais profundamente: acoita_cavalo (1), cordia_trichotoma (2), myracroduon_urundeuva (3). A (1) foi selecionada por causa de seu PCC de 1, que é o valor máximo de PCC. O (2) foi selecionado porque tem um PCC entre o segundo e terceiro quartil das classes no experimento e (3) foi selecionado porque tem o PCC mais baixo de todas as classes no experimento.

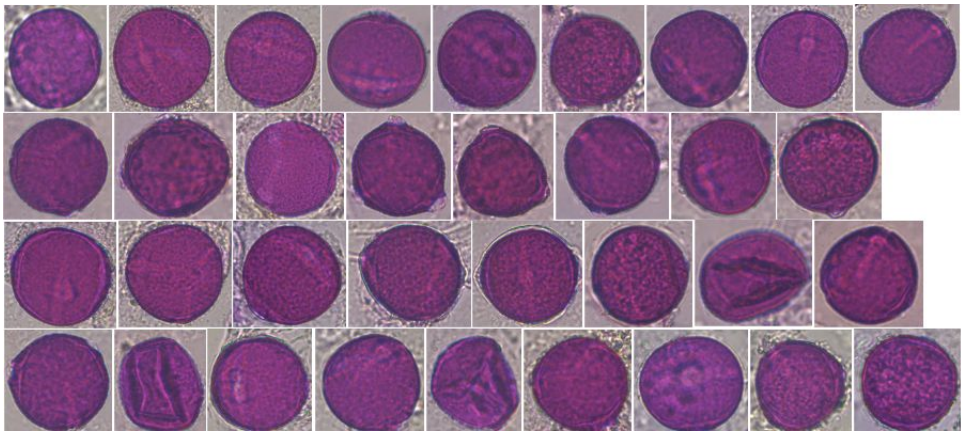

Figura 13: Imagens do pólen acoita cavalo (1).

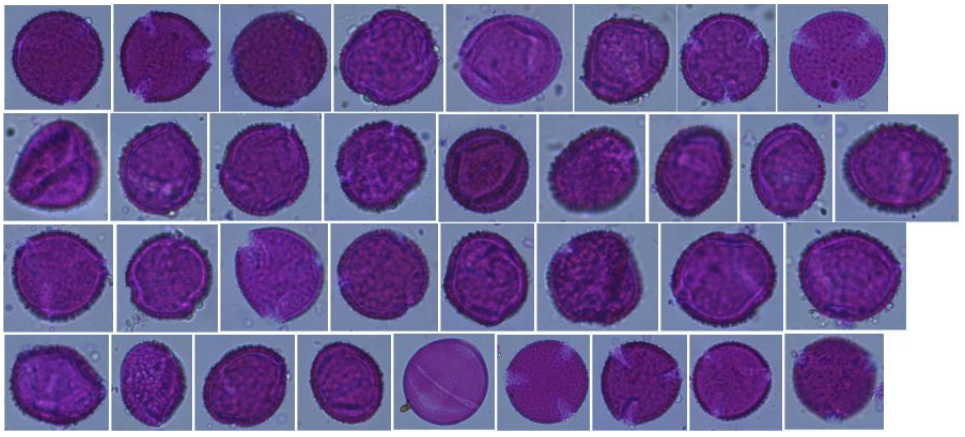

Figura 14: Imagens da cordia trichotoma (2).

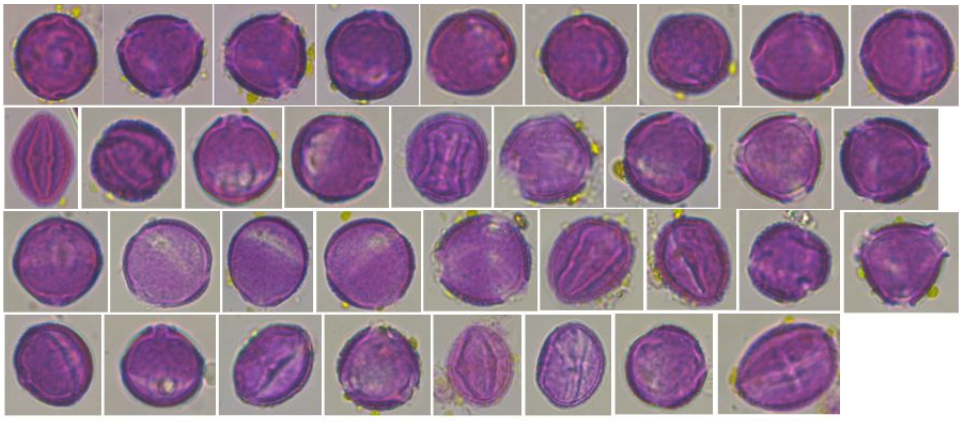

Figura 15: Imagens da myracroduon urundeuva (3).

Há de se levar em conta a forma e também as diferentes informações que cada *superpixel* pode levar em cada imagem porque o programa conta o número de *superpixels* parecidos. O que chama mais atenção entre estes 3 pólens é a diferença de consistência deles.

Pode-se observar como a (1) tem sempre a mesma forma e mesmo interior, tirando alguns pólens que mostram o buraco por dentro ou a "marca de tangerina" do pólen. Porém, perceba que há vários pólens com o tubo interno ou a "marca de tangerina", logo, o classificador sempre teve informações das várias visões do pólen para a classificação.

Enquanto isso, o (2) tem pólens em várias formas diferentes, inclusive a forma de estrela de 3 lados, vale ressaltar que muito outros pólens também têm esta forma, logo não é improvável que o classificador tenha se confundido com os outros pólens.

Agora, o (3) tem mais inconsistências que os outros 2, tem várias inconsistências de forma, inconsistência de presença de elementos (os pontos amarelos presentes em alguns deles), interiores diferentes e um elemento que parece ser de outras classes, o (3) claramente não tem informações o suficiente sobre todas as suas visões.

Uma solução proposta para pólens com PCC ruim como o (3) é o aumento de imagens de treino para o classificador ter mais informações sobre este pólen. Dizemos isso porque tanto o (1) quanto o (2) tem formas diferentes e interiores diferentes, mas como são poucas variações, o classificador não precisou de mais de 35 imagens para poder classificá-los competentemente.

6.1. Estatísticas dos Métodos de Classificação

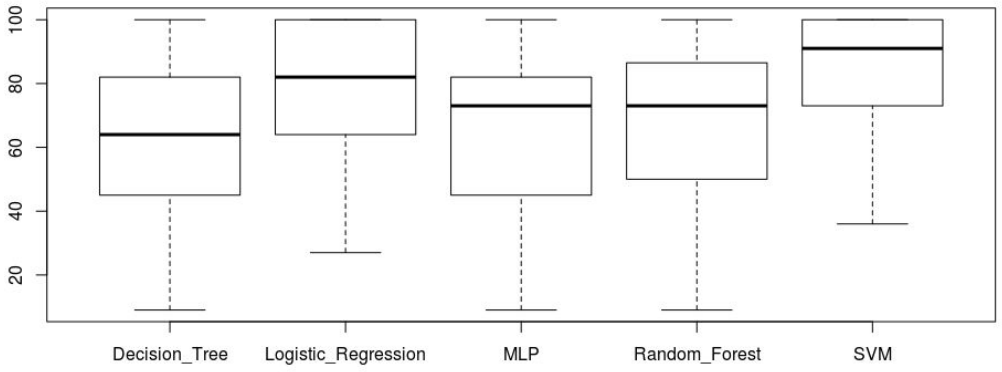

Figura 16: BoxPlot dos métodos de classificação.

Como mostrado no BoxPlot dos métodos de classificação, apresentado na Figura 16, o método SVM é o melhor método de classificação com a regressão logística como o segundo melhor método.

6.1.1. Teste de Friedman

Foi aplicado o teste de Friedman sobre os métodos de classificação para checar se não há diferença entre eles e o resultado de valor-p igual a 1.443e-15 prova que os métodos são bem diferentes entre si, dando mais confiança de que o SVM ou a regressão logística são os métodos superiores para este tipo de problema.

Tabela 5: Nível de significância entre os métodos de classificação.

Regressão Logística	Árvore de Decisão	2.57e-9
MLP	Árvore de Decisão	6.90e-1
Random Forest	Árvore de Decisão	5.12e-2
SVM	Árvore de Decisão	1.11e-15
MLP	Regressão Logística	5.55e-6
Random Forest	Regressão Logística	3.81e-3
SVM	Regressão Logística	2.67e-1
Random Forest	MLP	6.16e-1
SVM	MLP	1.93e-11
SVM	Random Forest	1.86e-7

Foi feito o pós-teste utilizando o teste post-hoc e os resultados estão apresentados acima. O pós-teste apresenta o nível de significância entre os métodos e embora os métodos de SVM e regressão logística serem entre os mais semelhantes do grupo, ainda pode-se dizer que nenhum método é semelhante, o que deixa o SVM ainda como o melhor método de classificação.

7. CONSIDERAÇÕES FINAIS

Temos de tomar cuidado com o fato de um histograma perder as informações como localização de *superpixels* e dependendo do jeito que um *superpixel* é formado, perdemos a forma da imagem. No entanto, mesmo assim, com taxas de acerto nos 80% pode-se dizer que histogramas de *superpixels* são uma ótima maneira de classificar imagens mais rapidamente pela simplicidade da informação. A melhor técnica de classificação neste trabalho é o SVM com *k* (número de centroides) sendo 85.

Ainda há várias técnicas de classificação a serem exploradas e vale ressaltar que a MLP tem vários parâmetros que podem ser modificados e testados, sendo que neste trabalho, foram usados apenas os parâmetros padrão.

Uma das grandes vantagens da técnica de histograma de superpixels é a perda da informação do posicionamento dos pixels. Isso é muito vantajoso porque não é necessário que se tenha as imagens de um objeto em todos os ângulos.

Ao mesmo tempo, a grande desvantagem desta técnica ainda é a extração de superpixel, foi usado o framework OpenCV para extrair todas os superpixels das imagens e o tempo de extração por imagem chega a 10 segundos, logo, um banco de imagens pequeno como o usado neste trabalho demora cerca de 6 horas para ser extraído. Esta é a nova fronteira para a tecnica de histograma de superpixels.

8. REFERÊNCIAS BIBLIOGRÁFICAS

BRYANT, M. V. **Forensic Palynology: Why it Works**. Palynology Laboratory, Texas A&M University. 2009. Disponível em: < http://projects.nfstc.org/trace/2009/presentations/3-bryant-palynology1.pdf >. Acesso em 04 abril de 2017.

GONÇALVES, Ariadne Barbosa. **Validação de Métodos Baseados em Visão Computacional para Automação da Identificação de Grãos de Pólen.** 2015. Dissertação (Mestrado) – Centro de Ciências Exatas e da Terra, Universidade Católica Dom Bosco, Campo Grande.

WRAY, Diana. **Vaughn Bryant Uses Pollen to Pinpoint Where a Victim has Been and Maybe Solve a Crime**. Houston Press, 8, jun, 2016. Disponível em: <http://www.houstonpress.com/news/vaughn-bryant-uses-pollen-to-pinpoint-where-a-victim-has-been-and-maybe-solve-a-crime-8519184> Acesso em 04 abril de 2017.

BALLARD, H. Dana; BROWN, M. Christopher. **Computer Vision**. Prentice-Hall, 1982. Disponível em: <http://homepages.inf.ed.ac.uk/rbf/BOOKS/BANDB/Ballard__D._and_Brown__C._M.__1982__Computer_Vision.pdf> Acesso em 02 junho de 2017.

PISTORI. **Etapas de um sistema de visão computacional**, 2013.

SALDANHA, M.; FREITAS, C. **Segmentação de Imagens Digitais: Uma Revisão**. 2008. Disponível em: <http://www.lac.inpe.br/cap/arquivos/pdf/P19.pdf>. Acesso em 02 junho de 2017.

LINARES, Oscar A. C. **Segmentação de imagens de alta dimensão por meio de algoritmos de detecção de comunidades e super pixels**, Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação, 2013.

SMITH, K. **SLIC Superpixels**. EPFL Technical Report 149300, June 2010. Disponível em: <http://www.kev-smith.com/papers/SLIC_Superpixels.pdf>. Acesso em 04 abril de 2017.

SEWELL, Martin. **Machine Learning.** Department of Computer Science, University College London. 2007.

LORENA, Ana Carolina; CARVALHO, André C. P. F. L. de. **Uma Introdução as Support Vector Machines**. 2007.

GIRARDELLO, A. D. **Um Estudo sobre o Uso de Máquinas de Vetores de Suporte em Problemas de Classificação**. 2010. Trabalho de Conclusão de Curso (TCC) - Centro de Ciências Exatas e Tecnológicas, Universidade Estadual do Oeste do Paraná, Cascavel.

RUMELHART, D. **Learning representations by back propagating errors** Nature vol 323 pag. 533. Out. 1986.

HOSMER, D. W. **Applied Logistic Regression**. EUA: Wiley-Interscience, 2000. p. 375.

SILVA, Gercina Gonçalves da. **Superpixel e Aprendizagem Supervisionada para a Identificação de Doenças da Soja em Imagens obtidas por Veículos Aéreos Não Tripulados.** 2017. Tese (Doutorado) – Centro de Ciências Exatas e da Terra, Universidade Católica Dom Bosco, Campo Grande.

JAIN, A.K. **Data clustering: 50 years beyond K-means**. Pattern Recognition Letters 31.8 (2010), Disponível em: <http://mlsurveys.s3.amazonaws.com/45.pdf>. Acesso em 04 abril de 2017.

BÉJAR, J. **K-means vs Mini Batch K-means: A comparison**. Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, [2013].

CHAPELLE, Olivier.; et al. **Semi-Supervised Learning**. The MIT Press, 2006.

LEE, H.C. **Introduction to Color Imaging Science**. Cambridge, 2005.

SANTOS, L. D. M. dos; MIKAMI, R.; VENDRAMIN, A. C. B. K.; KAESTNER, C. A. A.. **Procedimentos de Validação Cruzada em Mineração de Dados para ambiente de Computação Paralela**. ERAD 2009 — Caxias do Sul, 17 a 20 de março de 2009.

SCULLEY, D. **Web-Scale K-Means Clustering**. Tufts University, Boston, MA. Abril 2010. Disponível em: <*http://www.eecs.tufts.edu/~dsculley/papers/fastkmeans.pdf*> Acesso em: 05 out. 2017.

FEIZOLLAH, A. **Comparative Study of K-means and Mini Batch Kmeans Clustering Algorithms in Android Malware Detection Using Network Traffic Analysis.** University of Malaya, Kuala Lumpur, Malaysia. 2014. Disponível em: <https://www.researchgate.net/profile/Ali_Feizollah/publication/268386465_Comparative_Study_of_K-means_and_Mini_Batch_K-_means_Clustering_Algorithms_in_Android_Malware

_Detection_Using_Network_Traffic_Analysis/links/5469d5680cf2f5eb18052107.pdf>
Acesso em: 05 out. 2017.

MOHAMAD , I. B. Usman D. **Standardization and Its Effects on K-Means Clustering Algorithm.** Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310, UTM. Fev. 2013. Disponível em: <http://maxwellsci.com/print/rjaset/v6-3299-3303.pdf> Acesso em: 05 out. 2017.

PREPROCESSAMENTO. **Preprocessing data**. Disponível em: http://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-scaler Acesso em : 05 out. 2017.