Using syntactic methods and LSTM to the recognition of objects visual patterns

Gilberto Astolfi[1,2], Vanessa Aparecida de Moares Weber[3], Adair da Silva Oliveira Junior[1],
Geazy Vilharva Menezes[1], Nícolas Alessandro de Souza Belete[3], Everton Castelão Tetila[3], Hemerson Pistori[1,3]
[1]*College of Computing, Federal University of Mato Grosso do Sul (UFMS)*
[2]*Federal Institute of Education, Science and Technology of Mato Grosso do Sul (IFMS)*
[3]*Dom Bosco Catholic University (UCDB)*
gilberto.astolfi@ifms.edu.br

***Resumo*—In this paper, we have designed a new approach to represent and recognize objects visual patterns using syntactic methods. We capture relevant information from an object and associate them with symbols of an alphabet. After that, we derive a string from the object and input it to LSTM. The idea is to train LSTM with objects visual patterns encapsulated in the strings. We conducted an experiment using soybean crops aerial images captured by an Unmanned Aerial Vehicle (UAV), and we reached an average F-measure of 91%.**

***Index Terms*—Aerial images, precision crop protection, unmanned aerial vehicle (UAV), syntactic methods, LSTM.**

I. Introdução

Os métodos sintáticos são derivados das disciplinas de linguagens formais para reconhecimento de padrões visuais. Eles foram muito explorados em visão computacional na área de reconhecimento de padrões estruturais na década de setenta [1]. O uso de técnicas derivadas das linguagens formais é um campo clássico da visão computacional que pode ser traçado desde o início dos anos sessenta com tentativas de usar linguagens de cadeias de caracteres lineares para descrever imagens [2]. Recentemente, essas técnicas têm sido amplamente utilizadas em abordagens híbridas para representar relações hierárquicas de alto nível em imagens, como por exemplo, tarefas de compreensão de cenas [3], de reconhecimento de ações sequenciais [4] e análise de estruturas urbanas [5].

A principal questão na exploração de métodos sintáticos em visão computacional é representar os padrões visuais de imagens. Desde o início dos anos sessenta muitas abordagens foram propostas, especialmente após os anos dois mil [6], [7]. Nesse sentido, [8] propõem uma abordagem que combina detecção de pontos de interesse e inferência gramatical. Neste trabalho eles representam uma imagem como uma cadeia de caracteres e inferem gramáticas para cada classe de objetos a partir dessas cadeias. Adicionalmente, [9] introduz uma abordagem baseada em métodos sintáticos para representar ações e poses de pessoas. Neste, a imagem é dividida em retângulos com tamanhos e aspectos diferentes, organizados como primitivas em um grafo *And-Or*. Também usando grafos *And-Or*, [10] propõem um método para representar relacionamentos de contextos visuais de objetos. Ainda, [11] apresentou um modelo híbrido baseado em reconhecimento estrutural de padrões e SVM que permite que pessoas sejam detectadas em imagens. Finalmente, [12] usaram bag-of-visual-words para tarefas de compreensão de imagem.

Agradeçemos a Capes, CNPq e FUNDECT pelo apoio financeiro. A NVIDIA Corporation pela doação da GPU TITAN XP usada por esta pesquisa.

Neste trabalho, propomos uma abordagem híbrida, onde, primeiramente representamos os padrões visuais de objetos usando métodos sintáticos a fim de mapear um padrão visual de um objeto para uma cadeia de caracteres. Posteriormente, usamos uma Long Short-Term Memory (LSTM) para aprender os padrões visuais dos objetos encapsulados nas cadeias de caracteres. Experimentamos nossa abordagem usando imagens aéreas de lavouras de soja capturadas por um Veículo Aéreo Não Tripulado (VANT). O objetivo foi detectar problemas na lavoura da soja, como ervas daninhas e indícios de doenças. Comparamos nossa abordagem com quatro algoritmos rasos de aprendizagem de máquina e duas arquiteturas de rede neural profunda. Nossa abordagem apresentou resultados promissores, atingindo uma Medida-F média de 91%. A principal contribuição deste trabalho é mostrar a possibilidade de tratar padrões visuais de objetos de maneira sintática usando cadeias de caracteres, transformando o problema de reconhecimento de padrões em objetos, normalmente realizado por similaridade entre vetores de características ou tensores, em um problema de reconhecimento de padrões em cadeias de caracteres.

II. Fundamentação teórica

A. Reconhecimento sintático de padrões

A abordagem sintática de representação de padrões lida com os padrões sob uma perspectiva hierárquica e composicional. Um dado padrão complexo é composto por subpadrões mais simples, os quais são compostos por outros mais simples. No nível mais baixo dessa composicionalidade estão os padrões não divisíveis, que são chamados de primitivas [13]. Normalmente as primitivas são usadas para representar contornos, linhas ou texturas. Por outro lado, os padrões/subpadrões são usados para representar estruturas que dão forma a padrões visuais percebíveis. Esse modelo hierárquico e composicional de representação de padrões permite que primitivas e subpadrões sejam repetidos em diferentes padrões visuais, dando a possibilidade de representar vários padrões visuais a partir de um conjunto finito de dados. Além disso, ele provém uma descrição de como o padrão visual foi gerado. Em analogia a uma

linguagem, as primitivas são como as letras de um alfabeto, os subpadrões básicos são como palavras e padrões complexos, que dão forma a padrões visuais, como sentenças. Assim, uma linguagem pode ser projetada manualmente, quando aplicada a estrutura de padrões bem definida, ou inferida, quando se pretende aprendê-la a partir de dados de treinamento. Em ambos os casos, a linguagem tem o papel de representar exclusivamente uma classe de padrões ou de objetos [14].

A abordagem sintática de reconhecimento de padrões permite que o problema de reconhecimento de padrões em imagens, normalmente realizado por similaridade entre vetores de características ou tensores, possa ser tratada como um problema de reconhecimento de padrões em estrutura textual. Assim, um dado padrão visual passa a ser reconhecido como sendo de uma determinada classe por meio de *parsing* guiado por regras de sintaxe [13].

B. Scale Invariant Feature Transform (SIFT)

O Scale Invariant Feature Transform (SIFT) [15] é um algoritmo que detecta pontos salientes e estáveis em uma imagem. Esse ponto é chamado de ponto-chave e fornece um conjunto de características que descreve uma pequena região da imagem ao redor do ponto. A região descrita pelo ponto-chave é uma pequena região circular com uma orientação e invariante a rotação e escala. O ponto-chave é descrito por meio de quatro parâmetros: as coordenadas x e y do centro do ponto-chave, a escala (o raio da região), a orientação (um ângulo definido sobre o raio), e um descritor. O descritor de um ponto-chave é um vetor de 128 dimensões, calculado com base na magnitude e orientação do gradiente da imagem na região do ponto-chave.

C. Simple Linear Iterative Clustering (SLIC) Superpixels

Um superpixel é uma região em uma imagem formada por um conjunto de pixels que compartilham informações semelhantes de cores ou tons de cinza [16]. Os superpixels são obtidos por meio de algoritmos que tem como objetivo agrupar pixels semelhantes em regiões atômicas da imagem. Um desses algoritmos é o Simple Linear Iterative Clustering (SLIC) Superpixels proposto por [17]. O SLIC agrupa pixels com base na similaridade de cores e na proximidade espacial na imagem. O SLIC adapta o algoritmo k-means para realizar o agrupamento de pixels, por essa razão ele recebe como parâmetro um valor *k*, que corresponde a quantidade de superpixels que se deseja obter de uma dada imagem com tamanhos aproximadamente iguais. A Fig. 1 mostra um exemplo da aplicação do algoritmo SLIC em uma imagem com doença de lavoura da soja. Na primeira imagem o SLIC foi executado com o $k = 64$, na segunda com $k = 256$.

D. Long Short-Term Memory

Quando treinadas as Redes Neurais Recorrentes (RNN) [18] tomam como entrada não apenas o exemplo de dado atual, mas também informações que foram observadas anteriormente no tempo. O tempo aqui se refere a um tempo lógico que denota ordem numa sequência. Assim, as RNNs constroem

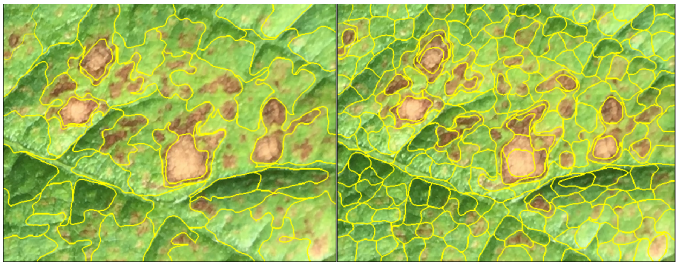

Figura 1. Imagem com doença da soja segmentada em 64 e 256 superpixels usando o algoritmo SLIC Superpixels.

seu modelo de reconhecimento de padrões combinando duas fontes de entrada de dados, o presente e o passado recente. Isso é possível devido às RNNs possuírem o *feedback loop*, que tem a função de se conectar a informações relevantes observadas no passado. O *feedback loop* faz com que a rede receba como uma de suas entradas a sua própria saída, criando um processo de retroalimentação de informação de maneira a construir ciclos. Cada ciclo armazena informações relevantes ao longo do tempo em estados ocultos mantidos pela rede. Dessa forma, a cada tempo t a rede não só armazena no seu estado oculto informações dos dados observados em t, como também recupera informações do estado oculto de $t-1$ que armazenou toda a informação relevante que aconteceu no passado. Assim, o estado oculto de t contém traços não apenas do estado oculto anterior $t-1$, mas também de todos aqueles que precederam $t-1$. Isso capacita a rede a criar correlações entre dados separados ao longo do tempo criando uma dependência entre eles de longo prazo, característica das RNNs popularmente chamadas de memória da rede.

A Long Short-Term Memory (LSTM) [19] é um tipo de RNN acrescida de uma célula de memória em cada tempo t, na qual se permite gravar e ler informações fora do fluxo normal criado pelos estados ocultos herdados da RNN. O acesso a escrita e leitura da célula é controlado por unidades chamadas portas. A decisão das portas de permitir acesso à célula é tomada com base em um conjunto de pesos que são calculados de acordo com o processo de aprendizagem da rede. A adição da célula de memória possibilita que a LSTM preserve um erro que pode ser retropropagado através do tempo e das camadas. A preservação do erro permite que a LSTM continue aprendendo por um período de tempo maior. Essa capacidade da LSTM corrige uma deficiência atribuída as RNNs comuns, de não conseguir propagar um erro constante por longos períodos de tempo [18].

Essa natureza da LSTM de manter informações de aprendizagem ao longo do tempo está intimamente relacionada ao reconhecimento de padrões em sequências de dados. Muitos trabalhos foram publicados explorando essa arquitetura natural da rede. [20] propôs uma abordagem para processamento de sequências de texto a fim de melhorar a tarefa de tradução automática. Com o mesmo propósito, tradução automática, [21] apresentou em seu trabalho uma abordagem que aprende a ordem de palavras em uma frase introduzindo dependência

entre elas. Há também trabalhos que tentam encontrar padrões em sequência genética [22] e [23]. Esses trabalhos são alguns exemplos que mostram a capacidade da LSTM de identificar padrões em sequência de dados.

III. Visão geral do método

O método consiste em uma estratégia de aprendizagem supervisionada que combina métodos sintáticos e a capacidade das LSTMs de aprender padrões em sequências. O objetivo é transformar padrões visuais de objetos em cadeias de caracteres que possam ser aprendidas por uma LSTM. O método é composto por fases. Primeiro, nós detectamos e representamos primitivas de objetos de maneira sintática. Para isso, nós detectamos os pontos-chave de um conjunto de objetos (superpixel) de treinamento usando o SIFT, e depois associamos a cada ponto-chave um símbolo de um alfabeto (veja seção III-A). Em seguida, derivamos uma cadeia de caracteres de cada objeto do conjunto de treinamento (veja seção III-B). Isso permite que descrevamos o padrão visual de cada objeto de maneira sintática. Finalmente, as cadeias de caracteres são inseridas em uma LSTM para aprendizagem (veja seção III-C) ou reconhecimento dos padrões visuais dos objetos (veja seção III-D) encapsulados nas cadeias de caracteres.

A. Identificando e representando primitivas

O objetivo nessa fase do método é identificar e representar primitivas de maneira sintática, isto é, associar primitivas de objetos a símbolos de um alfabeto.

Como apresentado na seção II-A, primitivas são padrões básicos e não divisíveis de objetos. Com base nessa definição, vamos considerar como primitivas no método proposto os pontos-chave de objetos detectados pelo SIFT.

Cada objeto (superpixel) de um dado conjunto de treinamento é computado pelo SIFT. Para cada ponto-chave detectado no objeto, como apresentado na seção II-B, o SIFT gera um vetor de característica contendo 128 valores e uma coordenada *xy*. A quantidade de pontos-chave detectados pelo SIFT em um objeto depende da variação de seu parâmetro *contrast*. Valor baixo de *contrast* implica detectar poucos pontos-chave.

Todos os vetores de características extraídos do conjunto de objetos de treinamento são agrupados, usando o *k*-means, e os *k* centros de agrupamentos resultantes formam um alfabeto de tamanho *k*. Por exemplo, quando os pontos-chave são agrupando usando *k* = 10, o alfabeto[1] resultante é o conjunto de símbolos $\Sigma = \{A, B, C, D, E, F, G, H, I, J\}$, onde cada centro de agrupamento é representado por um símbolo do alfabeto. Agora, cada ponto-chave pode ser representado por um símbolo do alfabeto correspondente ao seu centro de agrupamento mais próximo. Como estamos tratando pontos-chave como primitivas, com essa estratégia conseguimos representar as primitivas dos objetos de maneira sintática. A Fig. 2 mostra as primitivas de um objeto sendo representadas por símbolos de um alfabeto, isto é, de maneira sintática.

[1]Estamos usando símbolos do alfabeto português para representar os símbolos terminais do alfabeto. Mas, poderia ser usado qualquer outro conjunto de símbolos.

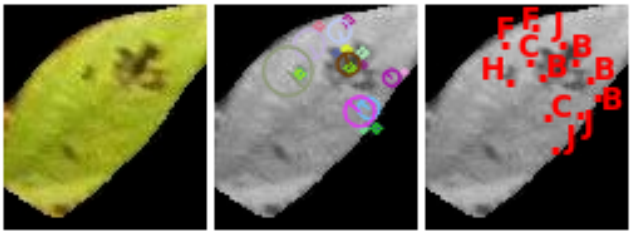

Figura 2. Da esquerda para a direita: Um exemplo de objeto (superpixel) de doença da soja; pontos-chave do objeto detectados pelo SIFT e; pontos-chave mapeados para um símbolo do alfabeto representando primitivas de maneira sintática.

B. Mapeando padrões visuais para cadeias de caracteres

O objetivo nessa fase do método é representar o padrão visual dos objetos de maneira sintática. A ideia é derivar uma cadeia de caractere de cada objeto. A estratégia consiste em concatenar os símbolos do alfabeto associados as primitivas de um dado objeto para formar uma cadeia de caracteres que represente o padrão visual do objeto.

Para cada objeto, inicialmente é criado o conjunto de primitivas P. Cada $p_i \in P$ é definida por uma tupla de dois elementos $p_i = (xy, \lambda)$, tal que xy é a localização espacial da primitiva no objeto e λ é o rótulo de p_i representado pelo símbolo do alfabeto associado a primitiva. Em seguida, é identificado o ponto central do objeto que chamaremos de p_{center}. Depois, é calculada a distância espacial a partir de p_{center} para todas as primitivas do conjunto P usando o elemento xy de cada primitiva. Com base na distância de p_{center} as primitivas são ordenadas em ordem crescente no conjunto P. Assim, $p_i \in P$ é a primeira primitiva do conjunto P e a primitiva mais próxima de p_{center}; $p_{i+1} \in P$ é a segunda primitiva do conjunto P e a segunda mais próxima de p_{center}; até p_n, a última primitiva do conjunto P e a mais distante de p_{center}. O próximo passo é concatenar os elementos λ de cada primitiva do conjunto P para formar uma cadeia de caractere. Então, o conjunto P é percorrido em ordem, concatenando o elemento λ de $\{p_i, p_{i+1} \ldots, p_n\} \in P$ para formar a cadeia de caractere s. Ao final do procedimento $|s| = |P|$. Para medir a distância de p_{center} para as primitivas nós usamos a distância Euclidiana.

Para exemplificar a formação da cadeia de caracteres a partir de primitivas de um dado objeto, tome como exemplo o objeto da Fig. 3 . Usaremos a terceira imagem, da esquerda para direita, para mostrar o processo de formação da cadeia de caracteres. As circunferências em laranja são usadas apenas para ilustrar as distâncias a partir do centro do objeto para cada primitiva. O p_{center} do objeto é o ponto amarelo no centro. A partir de p_{center} são calculadas as distâncias para todas as primitivas do objeto formando o conjunto $P = \{C, J, H, B, J, C, B, B, B, F, J, F\}$. P agora é percorrido, concatenando os símbolos, para formar a cadeia de caractere $s = CJHBJCBBBFJF$.

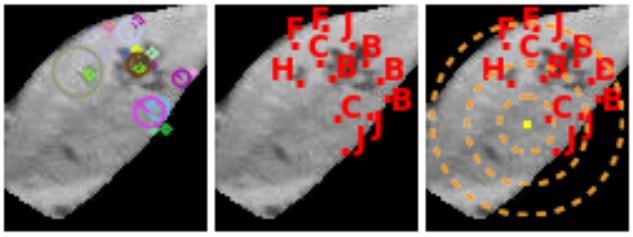

Figura 3. Da esquerda para a direita: pontos-chave do objeto detectados pelo SIFT; pontos-chave mapeados para um símbolo do alfabeto representando primitivas de maneira sintática e; símbolos das primitivas sendo alcançados a partir do centro da imagem para formar uma cadeia de caracteres, representando o padrão visual do objeto de maneira sintática.

Ao iniciar a composição da cadeia de caracteres a partir de um ponto central do objeto, evita-se que cadeias de caracteres diferentes sejam derivadas de um único objeto quando ele é rotacionado. Com essa estratégia conseguimos, por meio de regras de composição, representar o padrão visual de um dado objeto de maneira sintática, isto é, por meio de uma cadeia de caracteres.

C. Aprendendo padrões visuais de objetos

Nessa fase, o método já representou de maneira sintática, usando cadeias de caracteres, o padrão visual de cada objeto do conjunto de dados de treinamento. O próximo passo é treinar a LSTM com as cadeias de caracteres. As cadeias de caracteres são inseridas na LSTM na forma de tupla $(class, s)$, onde $class$ representa uma categoria e s um exemplo. Por exemplo, uma entrada para a LSTM poderia ser $(doenca\_soja,\ CJHBJCBBBFJF)$. Nesse caso, $doenca\_soja$ representa a classe *doença da soja* e $CJHBJCBBBFJF$ a cadeia de caractere que representa o padrão visual de um dado objeto *doença da soja*. Veja um exemplo gráfico na Fig. 4

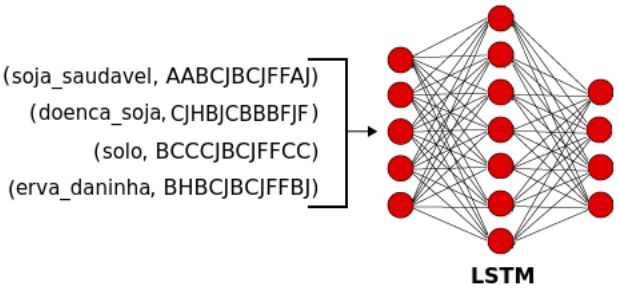

Figura 4. Exemplo de quatro padrões visuais de objetos distintos e de classes diferentes, representados por cadeias de caracteres, sendo inseridos no treinamento da LSTM.

D. Reconhecendo padrões visuais de objetos

O objetivo nesta seção é mostrar como é feito o reconhecimento dos padrões visuais dos objetos.

Como na fase de identificação e representação de primitivas, descrito na seção III-A, o conjunto de objetos, o qual se deseja fazer o reconhecimento de padrões, é computado pelo SIFT para detectar pontos-chave. Os vetores de características extraídos do conjunto de objetos não são usados para construir o alfabeto na fase de identificação e representação de primitivas (seção III-A). Eles são usados para mapear um símbolo do alfabeto, já definido, para cada ponto-chave. Para isso, é calculada a distância de um dado vetor de característica, de um determinado ponto-chave, para cada centro de agrupamento definido na fase de identificação e representação de primitivas. Tendo identificado o centro de agrupamento mais próximo do ponto-chave, mapeia-se o símbolo do alfabeto que representa o centro de agrupamento para o ponto-chave em questão. Esse procedimento é feito para todos os pontos-chave de todos os objetos. É importante dizer que o conjunto de objetos, o qual se deseja fazer o reconhecimento dos padrões, não é usado para construir o agrupamento na fase de identificação e representação de primitivas, consequentemente, ele também não é usado para criar o alfabeto. Isso significa que a predição do modelo LSTM não é influenciada pelos dados de treinamento.

Após mapear cada ponto-chave para um símbolo do alfabeto, ou seja, representar as primitivas de maneira sintática, a representação dos padrões visuais dos objetos por meio de cadeia de caracteres segue o mesmo procedimento definido na seção III-B. Finalmente, as cadeias de caracteres que representam os padrões visuais de cada objeto do conjunto de testes são submetidas à LSTM para predição.

IV. Experimentos

Nesta seção será apresentado um experimento com o método proposto em uma tarefa de classificação. O objetivo é detectar algum tipo de problema em lavouras de soja, como por exemplo, ervas daninhas ou doenças da soja. O método foi comparado com quatro algoritmos rasos e duas arquiteturas de aprendizagem profunda.

A. Materiais e métodos

O método foi avaliado usando um conjunto de dados composto por imagens de superpixels extraídos de lavouras de soja. O conjunto de dados foi construído a partir de imagens capturadas por um VANT, modelo DJI Phantom 3 Profissional com câmera Sony EXMOR 1/2.3", em um modo manual a uma altitude média de 4 metros acima do nível do solo. As imagens foram segmentadas usando o algoritmo SLIC, configurado para segmentar cada imagem em 80 segmentos. Foram classificados visualmente 400 segmentos de superpixels a partir das imagens em quatro classes: *soja saudável*, *soja com doença*, *solo* e *erva daninha*, cada uma contendo 100 exemplares de imagens de superpixel. Exemplos de imagens de superpixels do conjunto de dados são mostrados na Fig. 5.

O método foi avaliado usando 45 configurações diferentes de hiperparâmetros. O tamanho do alfabeto foi variado de 32 a 64, incrementando de 4 em 4. Antes de escolher essa faixa de valores para o tamanho do alfabeto, fizemos testes preliminares com o método variando o tamanho do alfabeto entre 8 a 256, dobrando o tamanho do alfabeto a cada teste (8, 16, 32, 64, 128 e 256). Os melhores resultados foram conseguidos com os tamanhos de alfabeto 32 e 64. Por essa razão decidimos variar

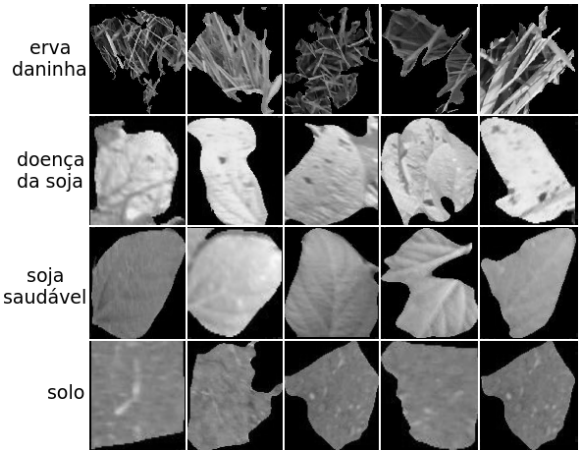

Figura 5. Exemplos de imagens do conjunto de dados. As classes são: erva daninha, soja com doença, soja saudável e solo. O conjunto de dados possui 400 imagens, 100 imagens para cada classe.

o tamanho do alfabeto entre 32 a 64. Cinco valores diferentes (0.04, 0.05, 0.06, 0.07 e 0.08) para o SIFT *contrast* foram testados. Esse hiperparâmetro determina o número de pontos-chave detectadas pelo SIFT. Quanto menor o valor, menos pontos-chave por imagem são detectados. Por fim, foi usado o valor padrão 100 para o tamanho da memória da LSTM. Como apresentado na seção II-D, a LSTM possui células de memória para preservar um erro que pode ser retropropagado através do tempo e das camadas da rede. Esse valor define a quantidade de células de memória que será usada pela LSTM.

O método foi comparado com algoritmos rasos de aprendizado KNN (k = 3, 5, 10), Árvores de Decisão (C4.5), Floresta Aleatória e SVM com os parâmetros padrão do WEKA 3.8.2. O método também foi comparado com as redes neurais profundas pré-treinadas ResNet-50 [24] e Xception [25], combinando os hiperparâmetros *batch size* = (4, 8, 16) e *transfer learning* com *fine-tuning rate* = (25, 50, 100), resultando em 9 configurações de testes diferentes para cada arquitetura.

Nos experimentos, foi usada a amostragem aleatória estratificada sobre o conjunto de imagens de superpixels, com 70% de exemplos de superpixels para treinamento e 30% para teste. Além disso, foram adotadas como métricas de avaliação a Precisão, Revocação, Acurácia e Medida-F.

B. Resultados e discussões

Na Tabela I são apresentados os melhores resultados obtidos entre as 45 diferentes configurações experimentadas no método proposto. É importante observar na Tabela I que os melhores resultados foram obtidos usando o valor de SIFT *contrast* de 0.04 a 0.06. Indicando que quanto maior o número de pontos-chave detectados nas imagens de superpixel, o método mostra melhores resultados. O resultado poderia ter sido melhor se o número de predições falso negativas fosse menor para as classes *soja saudável* e *solo*, quatro e sete respectivamente (ver Fig. 6). O que se destaca é que o método faz mais confusão entre essas duas classes. Em outras palavras, a classe *soja saudável* é confundida com a classe *solo* e vice-versa. A classe *solo* teve a maior incidência de predições falso negativas, sete no total. Nesse caso, objetos da classe *doença da soja* e *soja saudável* foram considerados como sendo da classe *solo* (Fig. 6)). No que diz respeito as predições verdadeiro positivas, as classes *erva daninha* e *doenças da soja* tiveram uma taxa de acerto de 100%. Esse desempenho pode ser explicado pela diferença visual notável entre os objetos das duas classes com as demais (veja Fig. 5).

Tabela I
MELHORES RESULTADOS ALCANÇADOS ENTRE 45 DIFERENTES CONFIGURAÇÕES EXPERIMENTADAS PELO MÉTODO.

Alfabeto	*contrast*	Precisão	Revocação	Medida-F	Acurácia
52	0.04	**90.8**%	**91.0**%	**91.0**%	**91.0**%
36	0.04	89.1%	90.0%	89.0%	89.0%
40	0.04	88.3%	88.0%	88.0%	88.0%
52	0.06	88.3%	88.0%	88.0%	88.0%
56	0.06	88.3%	88.0%	88.0%	88.0%
64	0.04	88.3%	89.0%	88.0%	88.0%
32	0.04	86.6%	87.0%	87.0%	86.0%
32	0.05	86.6%	87.0%	87.0%	87.0%
40	0.05	86.6%	88.0%	87.0%	87.0%
48	0.05	86.6%	87.0%	87.0%	87.0%
56	0.05	86.6%	87.0%	87.0%	87.0%
60	0.04	86.6%	86.0%	87.0%	86.0%

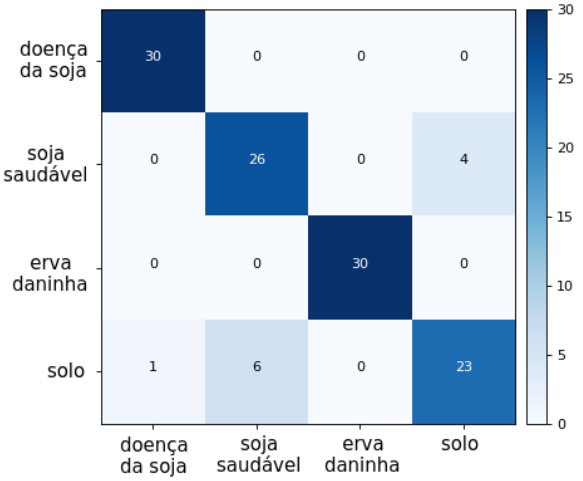

Figura 6. Matriz de confusão para o melhor resultado alcançado entre as 45 configurações diferentes experimentadas no método. Hiperparâmetros: Tamanho do alfabeto = 52 e SIFT *contrast* = 0.04.

Na Tabela II é apresentada uma comparação entre os algoritmos rasos de aprendizagem, o melhor resultado entre as arquiteturas de rede neural profunda e o melhor resultado obtido pelo método proposto. O melhor resultado alcançado pela ResNet-50 foi usando *batch size* = 4 e *transfer learning* com *fine-tuning rate* = 25, Xception foi usando *batch size* = 4 e *transfer learning* com *fine-tuning rate* = 100. O melhor resultado do método proposto foi usando os hiperparâmetros *tamanho do alfabeto* = 52 e *SIFT contrast* = 0.04. O método alcançou a maior média de Medida-F, o melhor resultado entre

Tabela II
COMPARAÇÃO ENTRE ALGORITMOS RASOS DE APRENDIZAGEM, ARQUITETURAS DE DEEP LEARNING E NOSSO MÉTODO PROPOSTO.

Algoritmos	Precisão	Recall	F-Measure	Acurácia
C4.5	83.4%	81.7%	82.2%	81.6%
Random Forests	85.0%	84.0%	85.0%	84.7%
SVM	**91.0**%	90.8%	90.9%	90.8%
KNN (k=10)	82.6%	78.3%	76.3%	78.3%
KNN (k=5)	81.7%	78.3%	75.6%	78.3%
KNN (k=3)	78.7%	79.2%	77.1%	79.1%
ResNet-50	86.0%	84.0%	84.0%	84.1%
Xception	84.0%	74.0%	68.0%	74.1%
Método proposto	90.8%	**91.0**%	**91.0**%	**91.0**%

todos os algoritmos testados. A SVM ficou em segundo lugar, com a maior média de Medida-F entre os outros algoritmos. Além disso, a SVM obteve uma leve melhor precisão em relação ao método proposto. A SVM e o método proposto tiveram resultados similares.

V. CONCLUSÕES

Nós propomos um novo método de reconhecimento sintático de padrões que se concentra em representar o padrão visual de objetos de maneira sintática. Para isso, informações relevantes de objetos, tratadas como primitivas, foram identificadas e representadas de maneira sintática. As primitivas foram relacionadas de maneira que pudessem encapsular o padrão visual do objeto em uma cadeia de caracteres. Por fim, as cadeias de caracteres foram introduzidas em uma LSTM para que ela pudesse aprender os padrões visuais dos objetos. Dessa forma, combinamos métodos sintáticos e a capacidade das LSTMs de aprender padrões em sequências. Com isso, transformamos o problema de reconhecimento de padrões em imagens, normalmente realizado por similaridade entre vetores de características ou tensores, em um problema de reconhecimento de padrões em cadeias de caracteres. O método foi testado usando um banco de imagens de superpixels construído a partir de imagens capturadas por um VANT. Nossa abordagem alcançou resultados promissores na tarefa de identificar problemas como ervas daninhas e doenças da soja. Como trabalhos futuros pretendemos evoluir o método para representar mais de um padrão visual por objeto, isto é, derivar mais de uma cadeia de caracteres por objeto. Dessa forma poderíamos representar padrões visuais de objetos usando sentenças. Ao representar um objeto usando sentenças, o problema de reconhecimento de padrões em imagens passa a ser um problema de reconhecimento de padrões em sentenças guiado por regras de sintaxe, contexto no qual as LSTMs apresentam resultados expressivos.

REFERÊNCIAS

[1] K. S. Fu, *Syntactic methods in pattern recognition*, ser. Mathematics in Science and Engineering. Academic Press, 1974, vol. 112.

[2] M. Eden, "On the formalization of handwriting," in *Structure of Language and its Mathematical Aspect.*, 1961, p. 83–88.

[3] J. Chua and P. F. Felzenszwalb, "Scene grammars, factor graphs, and belief propagation," *CoRR*, 2016.

[4] A. Fire and S. Zhu, "Inferring hidden statuses and actions in video by causal reasoning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 48–56.

[5] I. Demir, D. G. Aliaga, and B. Benes, "Procedural editing of 3d building point clouds," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2147–2155.

[6] R. W. D. Pedro, F. L. S. Nunes, and A. Machado-Lima, "Using grammars for pattern recognition in images: A systematic review," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 26:1–26:34, Nov. 2013.

[7] G. Chanda and F. Dellaert, "Grammatical methods in computer vision: An overview," College of Computing, Georgia Institute of Technology, Atlanta, GA, Tech. Rep. GIT-GVU-04-29, 2004.

[8] H. Pistori, A. Calway, and P. Flach, "A new strategy for applying grammatical inference to image classification problems," in *2013 IEEE International Conference on Industrial Technology (ICIT)*, Feb 2013, pp. 1032–1037.

[9] X. Song, T. Wu, Y. Jia, and S. C. Zhu, "Discriminatively trained and-or tree models for object detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3278–3285.

[10] M. Walton, D. Lange, and S.-C. Zhu, "Inferring context through scene understanding," in *AAAI Spring Symposium Series*, 2017.

[11] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Object detection with grammar models," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. USA: Curran Associates Inc., 2011, pp. 442–450.

[12] A. Foncubierta-Rodríguez, H. Müller, and A. Depeursinge, "From visual words to a visual grammar: using language modelling for image classification," *CoRR*, 2017.

[13] K.-S. Fu and A. Rosenfeld, "Pattern recognition and image processing," *IEEE Transactions on Computers*, vol. C-25, no. 12, pp. 1336–1346, Dec 1976.

[14] A. K. Jain, R. P. W. Duin, and Jianchang Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, Jan 2000.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans.on Pattern Anal. and Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[17] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," School of Computer and Communication Sciences and École Polytechnique Fédrale de Lausanne Joint Repor, Tech. Rep. EPFL Technical Report 149300, 2010.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[19] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *CoRR*, 2015.

[20] T. Luong, I. Sutskever, Q. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 11–19.

[21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.

[22] F. C. Cao R, S. M. Chan L, J. H, and C. Z., "Prolango: Protein function prediction using neural machine translation based on a recurrent neural network," *Molecules*, vol. 22, no. 10, pp. 176–218, 2017.

[23] F. Pouladi, H. Salehinejad, and A. M. Gilani, "Recurrent neural networks for sequential phenotype prediction in genomics," in *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, 2015, pp. 225–230.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1800–1807.