13º Congresso Interinstitucional de Iniciação Científica – CIIC 2019
30 e 31 de julho de 2019 – Campinas, São Paulo
ISBN: 978-85-7029-149-3

DETECÇÃO DE FRUTOS EM VITICULTURA UTILIZANDO REDES NEURAIS PROFUNDAS

Leonardo Lima de **Souza**[1]; Sandra **Avila**[2]; Thiago Teixeira **Santos**[3]

Nº 19603

RESUMO *– Neste trabalho investigamos técnicas de detecção de objetos por redes neurais aplicadas à detecção de frutos em viticultura. Desenvolvemos também a base de dados Embrapa WGISD, composta por imagens coletadas em Abril de 2017 e Abril de 2018 na Vinícola De Guaspari. Anotada manualmente, a base de dados possui 5 cultivares diferentes de uva: Syrah, Chardonnay, Cabernet Franc, Cabernet Sauvignon e Sauvignon Blanc, totalizando 4419 amostras de cachos de uva. Foram treinadas duas redes neurais convolutivas de arquiteturas, YOLOv2 e YOLOv3, para detecção e localização dos cachos nas imagens. Resultados quantitativos demonstraram precisão de até 88%, revocação de até 74%, e F1-Score de até 80% para YOLOv2 e precisão de até 92%, revocação de até 65% e F1-Score de até 76% para YOLOv3. Testes qualitativos demonstram que a rede YOLOv2 generaliza melhor para o conjunto de dados utilizado, e a rede YOLOv3 fornece uma localização melhor ajustada.*

Palavras-chaves: Detecção de frutos, Viticultura, Redes Neurais, Aprendizagem Profunda.

1 Autor, Bolsista CNPq (PIBIC): Graduação em Ciência da Computação, UNICAMP, Campinas-SP; leonardolimasza@gmail.com
2 Co-Orientadora, Pesquisadora e Docente no Instituto de Computação, UNICAMP, Campinas-SP; sandra@ic.unicamp.br.
3 Orientador: Pesquisador da Embrapa Informática Agropecuária, Campinas-SP; thiago.santos@embrapa.br.

13º Congresso Interinstitucional de Iniciação Científica – CIIC 2019
30 e 31 de julho de 2019 – Campinas, São Paulo
ISBN: 978-85-7029-149-3

ABSTRACT – *In this work we investigate object detection techniques by neural networks applied to the detection of fruit in viticulture. We also developed the Embrapa WGISD dataset, composed of images collected in April 2017 and April 2018 at the Guaspari Winery. Annotated manually, the dataset has 5 different cultivars of grape: Syrah, Chardonnay, Cabernet Franc, Cabernet Sauvignon and Sauvignon Blanc, totaling 4419 samples of grape bunches. Two convolutional neural networks of architectures, YOLOv2 and YOLOv3, were trained to detect and locate the bunches in the images. Quantitative results demonstrated up to 88% accuracy, up to 74% recall, and F1-Score up to 80% for YOLOv2 and accuracy up to 92%, recall up to 65% and F1-Score up to 76% for YOLOv3. Qualitative tests demonstrate that the YOLOv2 network generalizes better for the dataset used, and the YOLOv3 network provides a better adjusted location.*

Keywords: Fruit detection, Viticulture, Neural Networks, Deep Learning.

1 INTRODUÇÃO

O mapeamento e a estimativa de produção de frutos é importante na agricultura. A partir destas informações, o produtor pode localizar e classificar problemas na lavoura, determinar e aplicar medidas eficientes para o problema através de uma melhor utilização dos recursos disponíveis, por exemplo: fertilização, irrigação e poda. Técnicas acuradas de estimação de produção podem se tornar ferramentas importantes no planejamento e manejo em fruticultura.

Atualmente a estimativa de produção de frutos em pomares é comumente realizada de forma manual, por meio da contagem de amostras, tornando-a laboriosa. Esta abordagem pode possuir vieses humanos devidos à fadiga e à repetição exaustiva, os quais acarretam em estimativas imprecisas, prejudicando o planejamento do produtor e da indústria.

Abordagens recentes utilizam técnicas e soluções mais avançadas para realizar a contagem e mapeamento de forma econômica e eficaz, por exemplo técnicas de visão computacional e aprendizado de máquina baseadas em Redes Neurais Profundas (ou *Deep Learning*) (SA et al., 2016; BARGOTI et al., 2017). Trabalhos realizados anteriormente neste projeto investigaram a utilização de técnicas de aprendizado de máquina e Redes Neurais Convolucionais, com a arquitetura YOLO (*You Only Look Once*), em sua segunda versão YOLOv2 (REDMON. et. al.,

ISBN: 978-85-7029-149-3

2017). Os resultados obtidos na detecção de frutos foram melhores quando comparados à técnicas tradicionais (DOS SANTOS et al., 2017, 2018).

O objetivo principal do projeto de pesquisa, no qual este estudo faz parte, é predizer os cachos de uva das imagens capturadas por uma câmera embarcada em um sistema móvel. E para futuras aplicações em robótica agrícola, é essencial a utilização de um modelo que seja capaz de realizar as predições em tempo real, possibilitando uma detecção e localização dos cachos na imagem realizada por um sistema autônomo. Dessa forma a arquitetura YOLO é robusta o suficiente para realizarmos a predição de safra, um objetivo a curto prazo, não sendo necessária uma execução em tempo real, mas possibilita a sua utilização em tempo real quando necessário, em futuras aplicações do projeto.

As principais contribuições deste trabalho são: 1) desenvolvimento da base de dados Embrapa WGISD (*Wine Grape Instance Segmentation Dataset*), através da coleta de novos dados e anotações para *bounding boxes* e máscaras binárias; 2) avaliação de uma nova arquitetura YOLO, a YOLOv3 (REDMON et al., 2018), para a detecção de frutos em viticultura.

O texto está organizado da seguinte forma. Na Seção 2 apresentamos a metodologia utilizada e o conjunto de dados, processo de coleta, anotação, pré-processamento e aumentação, assim como a discussão sobre as arquiteturas YOLOv2 e YOLOv3. Na Seção 3 apresentamos os resultados obtidos pelos modelos, assim como as avaliações e métricas utilizadas. Por fim, na Seção 4 discutimos as descobertas obtidas através do trabalho, problemas e limitações encontradas, assim como possíveis áreas para melhorias e trabalhos futuros.

2 Materiais e Métodos

2.1 Aquisição da Base de Dados

A base de dados utilizada em trabalhos anteriores do projeto foi coletada previamente, na vinícola Guaspari (Espírito Santo do Pinhal, SP) em Abril de 2017, através da utilização de uma câmera digital SLR (Canon® EOS Rebel T3i), com lentes 18-55 milímetros configurada em modo automático (DOS SANTOS et al., 2017). A base inicialmente consistia de 59 imagens, apresentando frutos da variedade Syrah, com resolução de 1296 x 864 pixels.

Em Abril de 2018 foi realizada uma nova coleta, também na vinícola Guaspari, e foram obtidas imagens de diferentes variedades de uvas, sendo elas: Chardonnay, Cabernet Franc, Cabernet Sauvignon e Sauvignon Blanc (DOS SANTOS et al., 2018). Algumas imagens foram

ISBN: 978-85-7029-149-3

capturadas em diferentes dispositivos e com resoluções diferentes, sendo que 60 imagens de diferentes variedades, com exceção da Syrah, foram coletadas por um Moto Z2 Play em modo HDR, possuindo uma resolução de 4032 x 3024 pixels, e as demais 240 imagens de diferentes variedades foram coletadas pela mesma câmera SLR, possuindo resolução de 5184 x 3456 pixels.

Através da nova coleta de dados, em conjunto com os dados já disponíveis no projeto, possuímos um total de 300 imagens, com 5 variedades: Chardonnay, Cabernet Franc, Cabernet Sauvignon, Sauvignon Blanc e Syrah, identificadas na Figura 1.

Figura 1. Exemplos de imagens do conjunto de dados, sendo: (a) coletada anteriormente em Abril de 2017, (b), (c), (d), (e) coletadas em Abril de 2018.

2.2 Anotação da Base de Dados

Como se trata de um problema de detecção e localização de objetos — um problema de aprendizado supervisionado —, é necessário que o conjunto de dados forneça a localização dos cachos de uvas (*i.e.*, as posições em formato de *bounding boxes* — marcadores retangulares que localizam os objetos de interesse) para o modelo durante o treinamento. Neste trabalho modelamos o problema em apenas uma classe (cachos de uva), e portanto a base de dados contém amostras de *bounding boxes* representando apenas a localização dos cachos de uva.

A anotação foi realizada através da utilização da ferramenta *open source* LabelImg (TZUTA, 2015), ferramenta para anotação gráfica e rotulação de *bounding boxes* de objetos em imagens.

ISBN: 978-85-7029-149-3

Após a anotação obtivemos um número total de 4419 cachos. Na Figura 2 ilustramos alguns exemplos das *bounding boxes* anotadas (a) e (b).

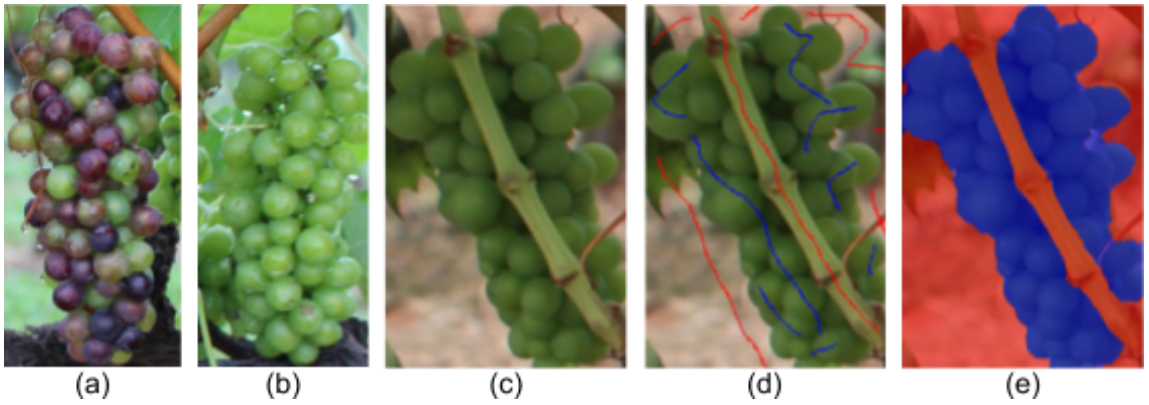


Figura 2. As imagens (a) e (b) representam exemplos de *bounding boxes* anotadas e (c), (d) e (e) representam o processo de anotação de máscaras utilizando a ferramenta produzida.

Também realizamos a anotação dos cachos de uva em máscaras binárias, uma vez que temos como objetivo futuro a utilização de modelos e arquiteturas para *instance segmentation* — segmentação de instância, onde cada pixel é identificado como um objeto e atribuído a uma instância (um cacho de uva neste caso). Segmentação de instâncias é interessante pois fornece informações espaciais mais precisas, delimitando acuradamente os frutos do restante da cena. Mask R-CNN (HE et al., 2017) é um exemplo de rede neural capaz de realizar segmentação de instâncias e que está em estudo no projeto. Porém a anotação em máscaras foi realizada somente para um conjunto menor de imagens, totalizando 2007 cachos. Esta anotação foi feita através de uma ferramenta produzida pelos autores deste trabalho, na qual utilizamos as *bounding boxes* anotadas anteriormente como entrada, e efetuamos a segmentação entre cachos, segundo plano e primeiro plano da imagem. O processo de segmentação consiste na utilização de uma técnica de *graph matching* gerando inicialmente um grafo relacional atribuído (*attributed relational graph* — ARG) para a imagem. A partir de alguns traços marcados pelo anotador, os quais o algoritmo produz um novo grafo, um emparelhamento (*matching*) dos grafos é computado, propagando a informação de marcação e identificando pixels nas classes uva e não-uva. Mais informações sobre a técnica e, uma discussão sobre o algoritmo podem ser encontradas em (NOMA et al., 2012) . Na Figura 2 mostramos um exemplo do processo de segmentação, (c), (d) e (e).

Para o treinamento e avaliação, das 300 imagens, a divisão utilizada foi 20% para o conjunto de teste (60 imagens) e 80% para o conjunto de treino e validação (240 imagens).

ISBN: 978-85-7029-149-3

2.3 Aumentação da Base de Dados

Apesar das arquiteturas de redes convolucionais profundas (CNNs) apresentarem um bom desempenho com algumas centenas de amostras, é demonstrado na literatura um aumento em desempenho através de técnicas de aumentação de dados (PEREZ et al., 2017). Assim, buscando uma melhora no desempenho, e também uma forma de inserir transformações semelhantes às possíveis variações em campo, utilizamos técnicas e transformações para aumentação da base de dados através da biblioteca *imgaug* (JUNG et al., 2018), uma biblioteca *open source* fornecida em linguagem Python.

Para efetuar a aumentação dos dados utilizamos as seguintes técnicas fornecidas pelo *imgaug*: *GaussianBlur, ConstrastNormalization, AdditiveGaussianNoise, Sharpen, DirectedEdge Detect, Dropout e CoarseDropout.* Efetuamos a aumentação aleatoriamente, onde a cada iteração todos os filtros possuem 50% de chance de serem aplicados, com exceção do *Dropout* e *CoarseDropout*, os quais possuem 25% de chance de acontecer. A aumentação foi feita somente para o conjunto de dados de treino que também possuem anotações em máscaras binárias. No final do processo, produzimos 20 novas imagens para cada, obtendo 1760 imagens aumentadas.

2.4 Arquitetura YOLO

O objetivo principal do nosso projeto de pesquisa é efetuar — em tempo real — a detecção e contagem de frutos, através de imagens fornecidas por uma câmera embarcada em um sistema móvel. Escolhemos utilizar modelos da arquitetura YOLO (REDMON et al., 2016), YOLOv2 e YOLOv3, na qual uma única rede neural realiza simultaneamente a detecção e classificação de objetos, fornecendo a localização e a classe do objeto, ambas predições de essenciais à realização de todo o processo de forma otimizada *"end-to-end"*. Ambas são velozes no processamento das imagens, processando-as em 45 FPS e 30 FPS respectivamente (*frames* por segundo).

Na estratégia YOLO, a rede utiliza características de toda a imagem para realizar a predição de uma *bounding box*, efetuando a predição para todas as classes simultaneamente. Divide-se a imagem de entrada em uma grade com *S* x *S células* (ver Figura 3), e caso o centro do objeto em questão esteja em uma célula da grade, esta célula se torna responsável pela detecção do objeto.

Cada célula realiza a predição de B *bounding boxes* e o valor de confiança para estas *bounding boxes*. Os valores de confiança indicam qual a confidência da *bounding box* conter o

ISBN: 978-85-7029-149-3

objeto, este valor é definido como: *Prob(Objeto)* x *IoU*, onde *Prob(Objeto)* é a probabilidade de existir um objeto dentro da *bounding box* e *IoU* é a interseção sobre a união, entre a *bounding box* verdadeira e a predita, na Figura 3 ilustramos o processo descrito. Caso nenhum objeto pertença a esta célula, o valor fornecido é zero, caso contrário queremos que o valor seja igual ao *IoU*.

As *bounding boxes* consistem de 5 valores: *x, y, w, h* e *confiança*, onde: *(x, y)* representam o centro da *bounding box* em relação às extremidades da célula, *w* e *h* são relativos às dimensões da imagem original, já a confiança é definida como discutido anteriormente. Como possuímos apenas uma classe uva, a saída fornecida é um tensor de dimensão 6 x 1.

As redes utilizadas YOLOv2 e YOLOv3 possuem algumas diferenças e características específicas, porém ambas seguem a abordagem descrita anteriormente. A seguir, iremos discutir a estrutura e características específicas de cada rede.

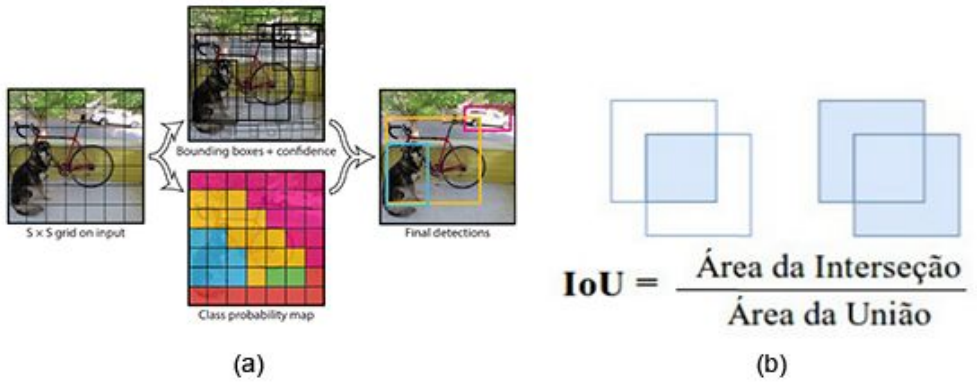


Figura 3. Ilustração referente ao fluxo da YOLO e cálculo do IoU, (a) e (b), respectivamente. Imagens retiradas de (REDMON et al., 2016).

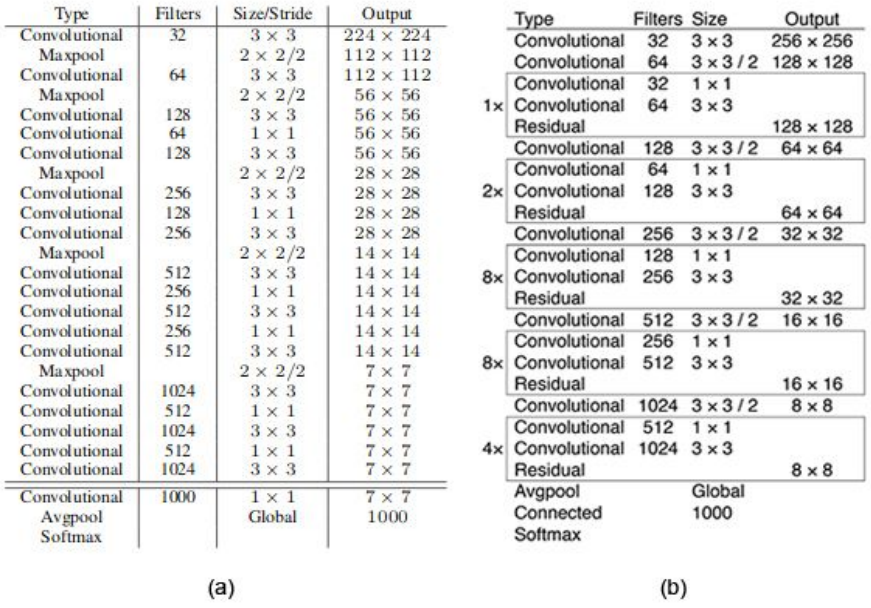
2.4.1 Arquitetura YOLOv2

A arquitetura YOLOv2 (REDMON et al., 2017) é baseada na GoogLeNet (SZEGEDY et al., 2015), que tem baixo tempo de execução. Porém o modelo final da YOLOv2 foi chamado de Darknet-19, possuindo 19 camadas convolucionais e 5 camadas de *maxpooling*, onde as camadas convolucionais são responsáveis pela extração das características da imagem (LECUN et al., 2015). Para utilização em detecção, a rede é alterada removendo a última camada convolucional e adicionando 3 camadas convolucionais 3 x 3, seguidas de uma camada convolucional 1 x 1 com o número de saídas necessárias para detecção. Uma representação da Darknet-19 é apresentada na Figura 4.

ISBN: 978-85-7029-149-3

2.4.2 Arquitetura YOLOv3

Para a YOLOv3 (REDMON et al., 2018), foi utilizada uma abordagem híbrida entre as redes YOLOv2, Darknet-19 e Redes Residuais (as quais evitam a sua degradação utilizando blocos residuais e saltos em conexões (HE et al., 2016)). Esta nova arquitetura utiliza camadas convolucionais consecutivas 3 x 3 e 1 x 1, possuindo ao todo 53 camadas convolucionais (Darknet-53). Na Figura 4 apresentamos uma representação da arquitetura da Darknet-53.



Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

(a)

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

(b)

Figura 4. Representação das redes Darknet-19 e Darknet-53, (a) e (b), respectivamente. Imagens retiradas de (REDMON et al., 2017) e (REDMON et al., 2018).

2.4.3 Treinamento e Avaliação

Os treinamentos de ambas as redes foram realizados utilizando o Darknet (REDMON, 2013), com as implementações da YOLOv2 e YOLOv3 disponíveis. Para ambos treinamentos, inicializamos as redes com pesos pré-treinados na base de dados ImageNet. O framework realiza automaticamente o *backup* dos pesos do treinamento a cada 100 iterações, até a iteração 900, e a partir disso o *backup* é feito a cada 10.000 iterações. Durante o treinamento foi analisado o comportamento do erro médio fornecido pelo *framework*. Através da análise dos erros, optamos por 170.000 iterações para YOLOv2 e 70.000 iterações para YOLOv3.

Para avaliar os modelos, utilizamos as métricas verdadeiro positivo, falso negativo e falso positivo, obtidas através das predições realizadas pelos modelos no conjunto de validação e teste.

ISBN: 978-85-7029-149-3

Consideramos verdadeiro positivo quando uma *bounding box* possui probabilidade para a classe uva acima de um *limiar* e uma *IoU* de 0.5, caso contrário é considerada um falso positivo. *Bounding boxes* não associadas a nenhum dos casos anteriores são consideradas como falso negativo. Através dos valores de verdadeiro positivo (Tp), falso positivo (Fp) e falso negativo (Fn), podemos utilizar as métricas de Precisão (P), Revocação (R) e F1-Score (F1) pelas fórmulas (Equação 1):

$$P = \frac{Tp}{Tp+Fp}, \quad R = \frac{Tp}{Tp+Fn}, \quad F1-Score = \frac{2 \times P \times R}{P+R} \qquad (1)$$

Efetuamos a avaliação das redes para as todas as combinações de limiares no intervalo [0,5 − 0,9] e pesos nos intervalos [10.000 − 170.000] para YOLOv2 e [10.000 − 70.000] para YOLOv3. Os limiares determinam se a predição é considerada aceitável para o conjunto final de respostas, caso o índice de confiança produzido seja maior ou igual ao limiar estabelecido.

3 Resultados e Discussão

As Tabelas 1, 2 e 3 mostram os resultados da avaliação dos modelos YOLOv2 e YOLOv3 nos conjuntos de validação e teste. A Tabela 1 é referente aos resultados dos melhores pesos e limiares para YOLOv2, a Tabela 2 é referente a YOLOv3 e a Tabela 3 ao resultado das duas melhores redes YOLOv2 e YOLOv3, no conjunto de teste de acordo com as Tabelas 1 e 2.

Analisando os resultados obtidos, ambas redes fornecem bons resultados, com uma diferença de 5 pontos percentuais entre os dois melhores F1-Score para o conjunto de validação. Além disso, ambas arquiteturas obtiveram precisão em torno dos 90% e revocação em torno dos 70%. No entanto, apesar dos resultados no conjunto de validação, os resultados obtidos no conjunto de teste (dados nunca vistos antes) tiveram quedas para ambas as redes, em torno de 8% (YOLOv2) e 13% (YOLOv3).

Ainda podemos observar que ambas redes possuem um número elevado de falso negativo no conjunto de teste, em comparação com o número total de 422 amostras, sendo 28,7% (YOLOv2) e 30% (YOLOv3). Isto ocorre pelo fato dos limiares utilizados serem elevados, pois nos baseamos no valor do F1-Score no conjunto de validação para eleger as melhores combinações. Isso indica que possíveis combinações com valores inferiores para validação podem generalizar melhor com imagens nunca vistas, como o conjunto de teste. A Figura 5 ilustra os resultados em uma imagem do conjunto de teste.

ISBN: 978-85-7029-149-3

Também podemos atribuir a queda no desempenho à quantidade de dados utilizada para o treinamento, pois embora tenhamos utilizado técnicas de aumentação, e obtido um grande conjunto de dados, efetuamos o treinamento utilizando somente o subconjunto de dados que também foram segmentados para podermos realizar uma futura comparação justa com a Mask R-CNN. Isso indica que possivelmente treinando em todo o conjunto de dados possíveis, podemos obter resultados ainda melhores, principalmente para a YOLOv3.

Como mencionado, podemos observar que a YOLOv2 possui um desempenho melhor que a YOLOv3. Isto ocorre pelo fato da arquitetura da YOLOv2 ser mais simples, o que para a quantidade de dados utilizada, possibilita convergir rapidamente com um aprendizado superior à YOLOv3. Por sua maior complexidade, em camadas e filtros, a YOLOv3 necessita de uma maior quantidade de dados para aprender todas as características necessárias, o que está de acordo com o apontado pela literatura em redes neurais convolutivas.

Tabela 1. Resultado da YOLOv2 para os melhores F1-Score no conjunto de validação.

Peso	Limiar	Verdadeiros Positivos	Falso Positivos	Falsos Negativos	Precisão	Revocação	F1
140.000	0,7	186	24	69	0,88	0,72	0,80
140.000	0,6	189	30	68	0,86	0,73	0,79
160.000	0,6	189	35	66	0,84	0,74	0,78
160.000	0,7	179	29	69	0,86	0,72	0,78

Tabela 2. Resultado da YOLOv3 para os melhores F1-Score no conjunto de validação.

Peso	Limiar	Verdadeiros Positivos	Falsos Positivos	Falsos Negativos	Precisão	Revocação	F1
30.000	0,8	136	11	74	0,92	0,64	0,76
30.000	0,7	144	13	77	0,91	0,65	0,76
30.000	0,5	149	17	77	0,89	0,65	0,76
20.000	0,5	158	23	78	0,87	0,66	0,75

Tabela 3. Resultado das YOLOv2 e YOLOv3 no conjunto de teste.

Rede	Peso	Limiar	Verdadeiros Positivos	Falsos Positivos	Falsos Negativos	Precisão	Revocação	F1
YOLOv2	140.000	0,7	233	44	121	0,84	0,65	0,73
YOLOv3	30.000	0,8	158	31	127	0,83	0,55	0,66

ISBN: 978-85-7029-149-3

4 Conclusão

Apresentamos neste trabalho uma abordagem para detecção de uvas utilizando redes neurais convolucionais com as arquiteturas YOLOv2 e YOLOv3. Também apresentamos a base de dados Embrapa WGISD, na qual coletamos novos dados de diferentes cultivares e aplicamos técnicas de anotação de *bounding boxes* e máscaras binárias, através da utilização da ferramenta desenvolvida em conjunto com técnicas de *graph matching*.

Ambas arquiteturas demonstraram bons resultados, sendo que no geral a YOLOv2 obteve uma performance melhor devido a sua característica e menor quantidade de camadas em comparação com a YOLOv3. Acreditamos que através da utilização da base de dados completa, iremos obter uma melhora no desempenho de ambas as redes.

Como trabalho futuro, buscamos investigar os resultados das arquiteturas em uma maior quantidade de dados anotados, e também realizar uma comparação com resultados obtidos pela arquitetura Mask R-CNN, pela utilização das máscaras já anotadas. Através de *instance segmentation* e *semantic segmentation* possibilitada pela Mask R-CNN, podemos determinar com uma maior precisão o volume de uvas em um pomar, informações as quais possuem um grande impacto na viticultura, possibilitando uma melhor tomada de decisão pelo produtor.

Figura 5. Resultados para as redes YOLOv2, YOLOv3 da Tabela 3, imagens (a) e (b), respectivamente.

5 Agradecimentos

Agradecemos ao PIBIC/CNPq pela bolsa concedida (#125044/2018-6) e Embrapa pela oportunidade de desenvolvimento do projeto, assim como ao Programa de Concessão de GPUs da

ISBN: 978-85-7029-149-3

NVIDIA (NVIDIA's GPU Grant Program) por ceder uma GPU Titan X, e também pelos orientadores Thiago e Sandra pelos conhecimentos e experiências compartilhadas.

6 REFERÊNCIAS

BARGOTI, Suchet; UNDERWOOD, James. Deep fruit detection in orchards. In: **IEEE International Conference on Robotics and Automation (ICRA)**. 2017. p. 3626-3633.

DOS SANTOS, Andreza A.; SANTOS, Thiago. T. Detecção de frutos em campo por aprendizado de máquina. In: **Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)**. In: Congresso Interinstitucional de Iniciação Científica, 11., Campinas, 2017.

DOS SANTOS, Andreza A.; AVILA, Sandra; SANTOS, Thiago T. Detecção automática de uvas e folhas em viticultura com uma rede neural YOLOv2. In: **Embrapa Informática Agropecuária-Artigo em Anais de Congresso (ALICE)**. In: Congresso Interinstitucional de Iniciação Científica, 12., Campinas, 2018.

HE, Kaiming et al. Deep residual learning for image recognition. In: **IEEE Conference on Computer Vision and Pattern Recognition**. 2016. p. 770-778.

HE, Kaiming et al. Mask R-CNN. In: **IEEE International Conference on Computer Vision**. 2017. p. 2961-2969.

JUNG, Alexander. Imgaug. **http://imgaug.readthedocs.io/en/latest/**. Acessado em: 12 jun. 2018.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, v. 521, n. 7553, p. 436, 2015.

NOMA, Alexandre et al. Interactive image segmentation by matching attributed relational graphs. **Pattern Recognition**, v. 45, n. 3, p. 1159-1179, 2012.

PEREZ, Luis; WANG, Jason. The effectiveness of data augmentation in image classification using deep learning. **arXiv:1712.04621**, 2017.

REDMON, Joseph. Darknet: Open source neural networks in c. 2013.

REDMON, Joseph et al. You only look once: Unified, real-time object detection. In: **IEEE Conference on Computer Vision and Pattern Recognition**. 2016. p. 779-788.

REDMON, Joseph; FARHADI, Ali. YOLO9000: better, faster, stronger. In: **IEEE on Computer Vision and Pattern Recognition**. 2017. p. 7263-7271.

REDMON, Joseph; FARHADI, Ali. YOLOv3: An incremental improvement. **arXiv:1804.02767**, 2018.

SA, Inkyu et al. Deepfruits: A fruit detection system using deep neural networks. **Sensors**, v. 16, p. 1222, 2016.

SZEGEDY, Christian et al. Going deeper with convolutions. In: **IEEE on Computer Vision and Pattern Recognition**. 2015. p. 1-9.

TZUTA, Lin, LabelImg. Git code (2015). **https://github.com/tzutalin/labelImg**. Acessado em: 24 jul. 2019.