Inception-v4, Inception-ResNet e o impacto das conexões residuais na aprendizagem

Christian Szegedy
Google Inc.
1600 Amphitheatre Pkwy, Mountain View, CA
szegedy@google.com

Sergei Ioffe
sioffe@google.com

Vincent Vanhoucke
vanhoucke@google.com

Alex Alemi
alemi@google.com

Abstrato

Redes convolucionais muito profundas têm sido centrais para os maiores avanços no desempenho de reconhecimento de imagem em anos recentes. Um exemplo é a arquitetura Inception que demonstrou alcançar um desempenho muito bom com um custo computacional relativamente baixo. Recentemente, a introdução de conexões residuais em conjunto com uma arquitetura mais tradicional rendeu desempenho de última geração no desafio ILSVRC de 2015; seu desempenho foi semelhante para a rede Inception-v3 de última geração. Isso aumenta a questão de saber se há algum benefício em combinar a arquitetura Inception com conexões residuais. Aqui damos evidências empíricas claras de que o treinamento com resíduos conexões acelera o treinamento de redes Inception significativamente. Há também algumas evidências de que redes residuais de Inception superam o desempenho de redes Inception igualmente caras. redes sem conexões residuais por uma pequena margem. Nós também apresentam várias novas arquiteturas simplificadas para ambos redes de iniciação residuais e não residuais. Essas variações melhoram o desempenho do reconhecimento de quadro único em significativamente a tarefa de classificação do ILSVRC 2012. Demonstramos ainda como o escalonamento de ativação adequado estabiliza a formação de redes iniciais residuais muito amplas. Com um conjunto de três resíduos e um Inception-v4, nós atingir 3,08% de erro top-5 no conjunto de teste do ImageNet desafio de classificação (CLS).

1. Introdução

Desde a competição ImageNet de 2012 [11] vencedora da inscrição de Krizhevsky et al [8], sua rede "AlexNet" tem foi aplicado com sucesso a uma variedade maior de computadores tarefas de visão, por exemplo, detecção de objetos [4], segmentação [10], estimativa de pose humana [17], classificação de vídeo ção [7], rastreamento de objetos [18] e superresolução [3]. Esses exemplos são apenas algumas de todas as aplicações para as quais redes convolucionais foram aplicadas com muito sucesso desde então.

Neste trabalho estudamos a combinação dos dois mais ideias recentes: Conexões residuais introduzidas por He et al. em [5] e a última versão revisada da arquitetura Inception [15]. Em [5], argumenta-se que as conexões residuais são de importância inerente para treinar arquiteturas muito profundas. Como as redes Inception tendem a ser muito profundas, é natural substituir o estágio de concatenação de filtros do Inception arquitetura com conexões residuais. Isso permitiria Início para colher todos os benefícios da abordagem residual mantendo sua eficiência computacional.

Além de uma integração direta, também estudamos se o próprio Inception pode se tornar mais eficiente através de tornando-o mais profundo e amplo. Para isso, projetamos uma nova versão chamada Inception-v4 que possui uma arquitetura simplificada mais uniforme e mais módulos de iniciação do que Inception-v3. Historicamente, o Inception-v3 herdou muita da bagagem das encarnações anteriores. As restrições técnicas vieram principalmente da necessidade de particionamento o modelo para treinamento distribuído usando DistBelief [2]. Agora, depois de migrar nossa configuração de treinamento para o TensorFlow [1], estes restrições foram levantadas, o que nos permitiu simplificar significativamente a arquitetura. Os detalhes disso simplificado arquitetura são descritas na Seção 3.

Neste relatório, compararemos os dois Inception puros variantes, Inception-v3 e v4, com versões híbridas Inception-ResNet igualmente caras. É certo que esses modelos foram escolhidos de uma forma um tanto ad hoc com o a principal restrição é que os parâmetros e a complexidade computacional dos modelos devem ser um tanto semelhantes ao custo dos modelos não residuais. Na verdade, testamos variantes Inception-ResNet maiores e mais amplas e tiveram desempenho muito semelhante no desafio de classificação ImageNet.

conjunto de dados longo [11].

O último experimento relatado aqui é uma avaliação de um conjunto de todos os modelos de melhor desempenho apresentados aqui. Como ficou evidente que tanto o Inception-v4 quanto o Inception-ResNet-v2 tiveram um desempenho semelhante, superando o desempenho de quadro único de última geração no conjunto de dados de validação do ImageNet, queríamos ver como uma combinação desses impulsiona o estado da arte neste conjunto de dados bem estudado. Surpreendentemente, descobrimos que os ganhos no desempenho de quadro único não se traduzem em ganhos igualmente grandes no desempenho do conjunto. No entanto, ainda nos permite relatar 3,1% de erro top-5 no conjunto de validação com quatro modelos reunidos, estabelecendo um novo estado da arte, até onde sabemos.

Na última seção, estudamos algumas das classificações fracassos e concluir que o conjunto ainda não atingiu o ruído do rótulo das anotações neste conjunto de dados e não ainda há espaço para melhorias nas previsões.

2. Trabalho relacionado

As redes convolucionais tornaram-se populares em grandes escalar tarefas de reconhecimento de imagem após Krizhevsky et al. [8]. Alguns dos próximos marcos importantes foram Network-in-network [9] de Lin et al., VGGNet [12] de Simonyan et al. e GoogLeNet (Inception-v1) [14] por Szegedy et al.

A conexão residual foi introduzida por He et al. em [5] nos quais eles fornecem evidências teóricas e práticas convincentes das vantagens da utilização da fusão aditiva de sinais tanto para reconhecimento de imagem, quanto especialmente para reconhecimento de objetos detecção. Os autores argumentam que as conexões residuais são inerentemente necessário para treinar convoluções muito profundas modelos. Nossas descobertas não parecem apoiar essa visão, pelo menos menos para reconhecimento de imagem. No entanto, pode exigir mais pontos de medição com arquiteturas mais profundas para entender a verdadeira extensão dos aspectos benéficos oferecidos pelas conexões residuais. Na seção experimental demonstramos que não é muito difícil treinar redes competitivas muito profundas sem utilizar conexões residuais. No entanto, o o uso de conexões residuais parece melhorar o treinamento velocidade muito, o que por si só é um grande argumento para seu uso.

A arquitetura convolucional profunda Inception foi introduzida em [14] e foi chamada de GoogLeNet ou Inception-v1 em nossa exposição. Mais tarde, a arquitetura Inception foi refinada de várias maneiras, primeiro pela introdução da normalização em lote [6] (Inception-v2) por Ioffe et al. Mais tarde a arquitetura foi melhorado por idéias adicionais de fatoração no terceiro iteração [15] que será referida como Inception-v3 em este relatório.

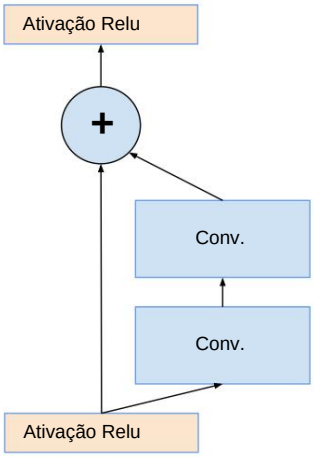

Figura 1. Conexões residuais introduzidas em He et al. [5].

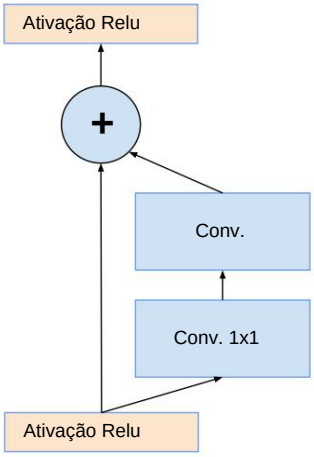

Figura 2. Versão otimizada das conexões ResNet por [5] para blindagem computação.

3. Escolhas arquitetônicas

3.1. Blocos de início puro

Nossos modelos Inception mais antigos costumavam ser treinados de maneira particionada, onde cada réplica era particionada em um múltiplas sub-redes para poder acomodar todo o modelo na memória. No entanto, a arquitetura Inception é altamente ajustável, o que significa que há muitas possibilidades mudanças no número de filtros nas várias camadas que não afetam a qualidade da rede totalmente treinada. Em para otimizar a velocidade de treinamento, costumávamos ajustar o tamanhos de camada cuidadosamente, a fim de equilibrar a computação entre as várias sub-redes do modelo. Em contrapartida, com o introdução do TensorFlow, nossos modelos mais recentes podem ser treinado sem particionar as réplicas. Isso está habilitado em parte por otimizações recentes de memória usadas por retropropagação, alcançadas considerando cuidadosamente quais tensores são necessário para cálculo de gradiente e estruturação do cálculo

para reduzir o número de tais tensores. Historicamente, temos sido relativamente conservadores quanto à mudança das escolhas arquitetônicas e restringimos nossos experimentos a vários componentes de rede isolados, enquanto mantemos o resto da rede estável. A não simplificação das escolhas anteriores resultou em redes que pareciam mais complicadas do que precisavam ser. Em nossos experimentos mais recentes, para o Inception-v4, decidimos nos livrar dessa bagagem desnecessária e fizemos escolhas uniformes para os blocos do Inception para cada tamanho de grade. Consulte a Figura 9 para a estrutura em grande escala da rede Inception-v4 e as Figuras 3, 4, 5, 6, 7 e 8 para a estrutura detalhada de seus componentes. Todas as convoluções não marcadas com "V" nas figuras são preenchidas da mesma forma, o que significa que sua grade de saída corresponde ao tamanho de sua entrada. As convoluções marcadas com "V" são preenchidas de forma válida, o que significa que o patch de entrada de cada unidade está totalmente contido na camada anterior e o tamanho da grade do mapa de ativação de saída é reduzido de acordo.

3.2. Blocos de Iniciação Residuais

Para as versões residuais das redes Inception, usamos blocos Inception mais baratos que o Inception original.
Cada bloco Inception é seguido por uma camada de expansão de filtro (convolução 1 × 1 sem ativação) que é usada para aumentar a dimensionalidade do banco de filtros antes da adição para corresponder à profundidade da entrada. Isto é necessário para compensar a redução de dimensionalidade induzida pelo bloco Inception.

Tentamos várias versões da versão residual do In-ception. Apenas dois deles são detalhados aqui. O primeiro "Inception-ResNet-v1" corresponde aproximadamente ao custo computacional do Inception-v3, enquanto "Inception-ResNet-v2" corresponde ao custo bruto da rede Inception-v4 recentemente introduzida. Veja a Figura 15 para a estrutura em grande escala de ambas as variantes.
(No entanto, o tempo de passo do Inception-v4 provou ser significativamente mais lento na prática, provavelmente devido ao maior número de camadas.)

Outra pequena diferença técnica entre nossas variantes residuais e não residuais do Inception é que, no caso do Inception-ResNet, usamos a normalização em lote apenas no topo das camadas tradicionais, mas não no topo dos somatórios. É razoável esperar que um uso completo da normalização em lote seja vantajoso, mas queríamos manter cada réplica do modelo treinável em uma única GPU. Descobriu-se que o consumo de memória de camadas com grande tamanho de ativação estava consumindo uma quantidade desproporcional de memória da GPU. Ao omitir a normalização em lote no topo dessas camadas, conseguimos aumentar substancialmente o número geral de blocos Inception. Esperamos que, com uma melhor utilização dos recursos computacionais, essa compensação se torne desnecessária.

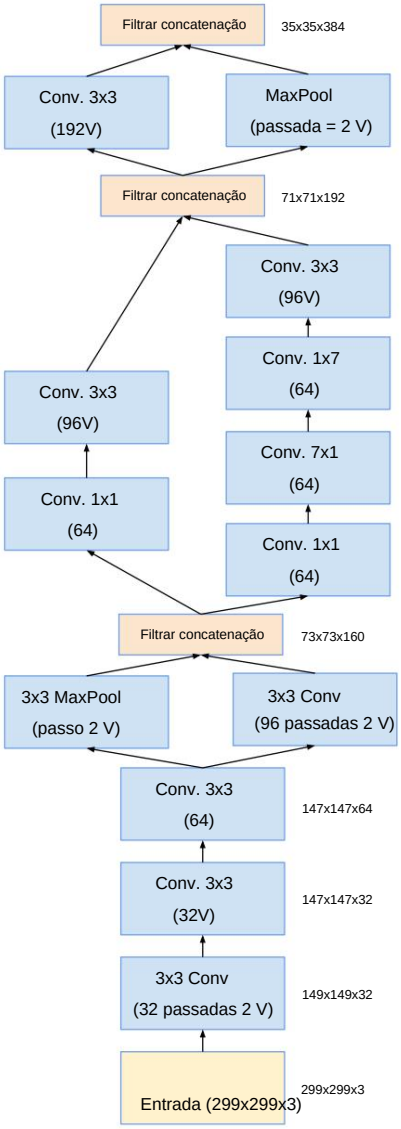

Figura 3. O esquema para o tronco das redes Inception-v4 e Inception-ResNet-v2 puras. Esta é a parte de entrada dessas redes. Cf. Figuras 9 e 15

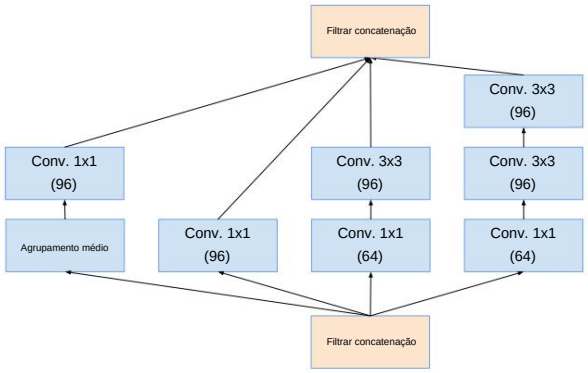

Figura 4. O esquema para módulos de grade 35 × 35 da rede Inception-v4 pura. Este é o bloco Inception-A da Figura 9.

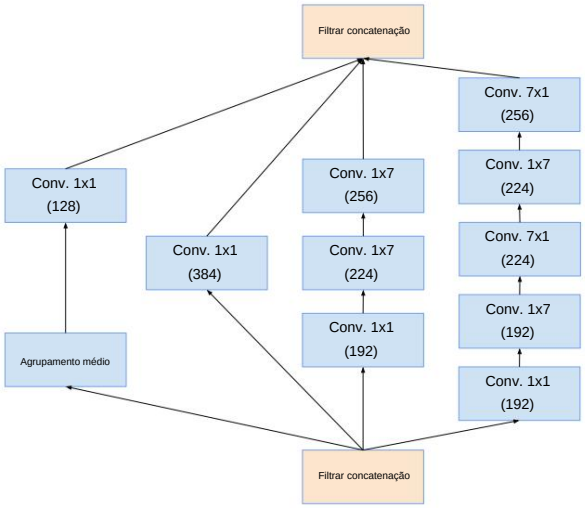

Figura 5. O esquema para módulos de grade 17 × 17 da rede Inception-v4 pura. Este é o bloco Inception-B da Figura 9.

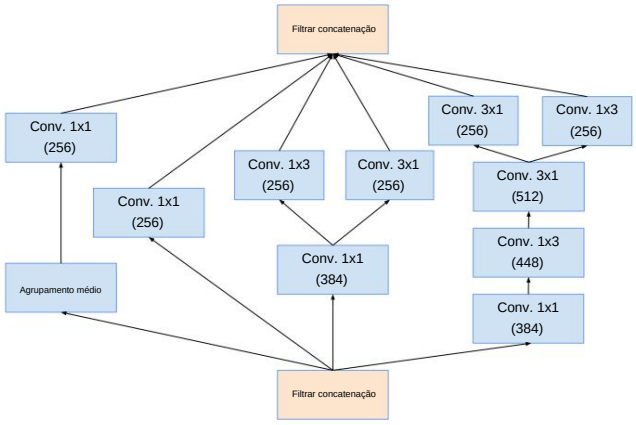

Figura 6. O esquema para módulos de grade 8×8 da rede Inception-v4 pura. Este é o bloco Inception-C da Figura 9.

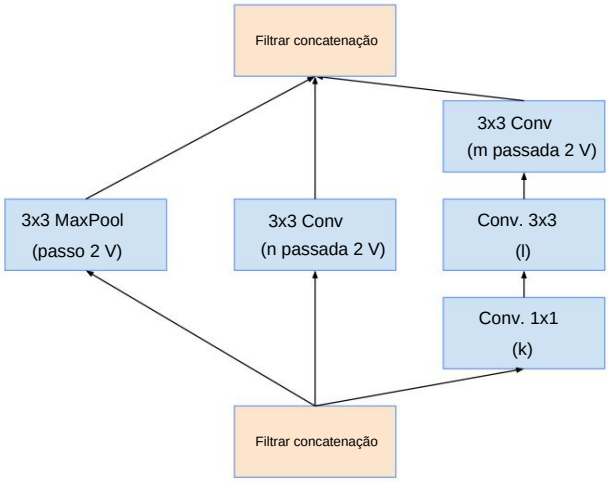

Figura 7. Esquema para módulo de redução de 35 × 35 a 17 × 17. Diferentes variantes desses blocos (com vários números de filtros) são usadas na Figura 9 e 15 em cada uma das novas variantes do Inception (-v4, - ResNet-v1, -ResNet-v2) apresentadas neste artigo. Os números k, l, m, n representam tamanhos de bancos de filtros que podem ser consultados na Tabela 1.

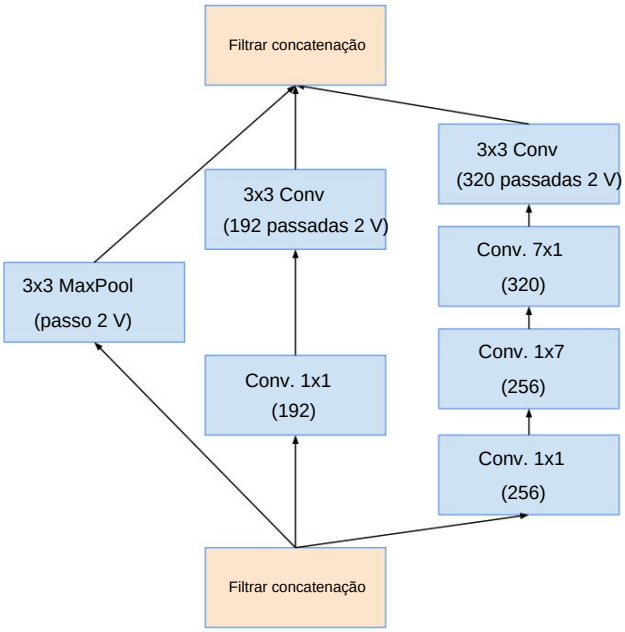

Figura 8. Esquema para módulo de redução de grade 17 × 17 a 8 × 8. Este é o módulo de redução usado pela rede Inception-v4 pura na Figura 9.

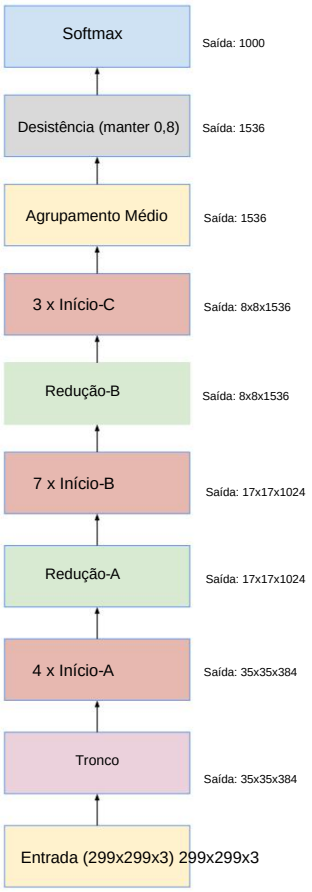

Figura 9. O esquema geral da rede Inception-v4. Para os módulos detalhados, consulte as Figuras 3, 4, 5, 6, 7 e 8 para a estrutura detalhada dos vários componentes.

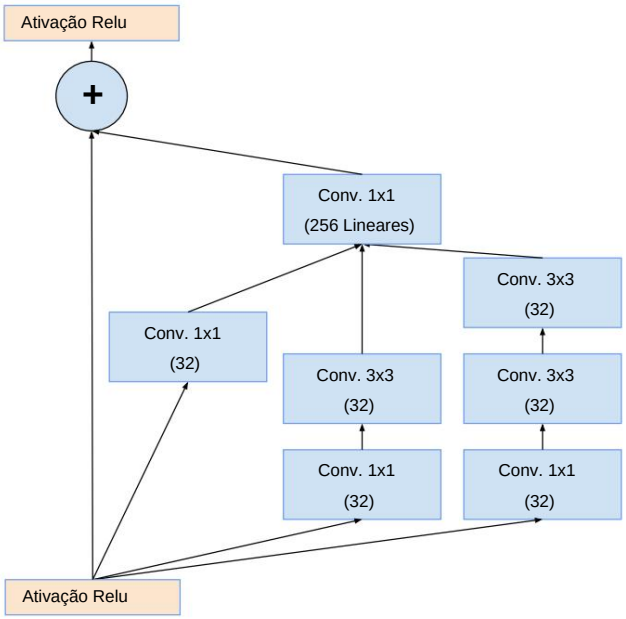

Figura 10. Esquema para módulo de grade 35 × 35 (Inception-ResNet-A) da rede Inception-ResNet-v1.

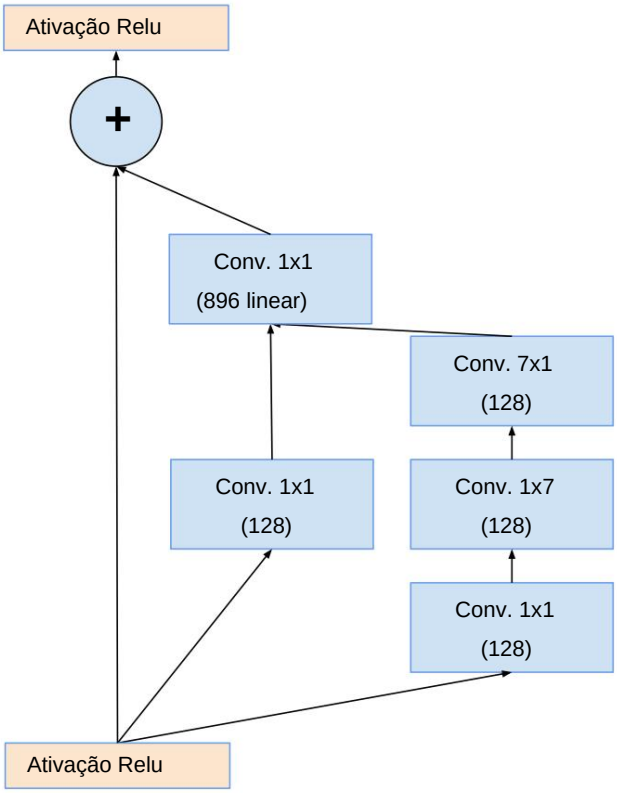

Figura 11. O esquema para o módulo de grade 17 × 17 (Inception-ResNet-B) da rede Inception-ResNet-v1.

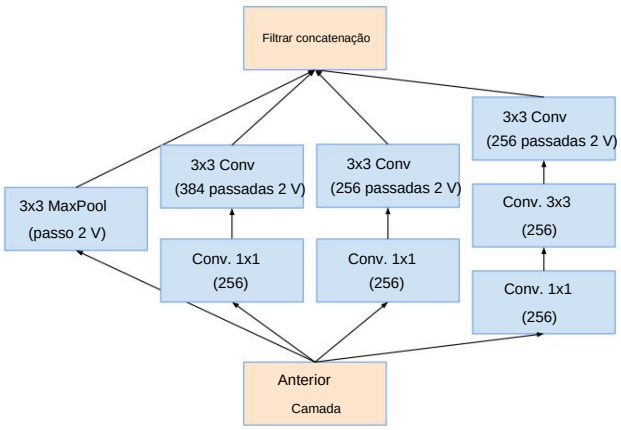

Figura 12. Módulo de redução de grade “Redução-B” 17×17 a 8×8. Este módulo é usado pela rede menor Inception-ResNet-v1 na Figura 15.

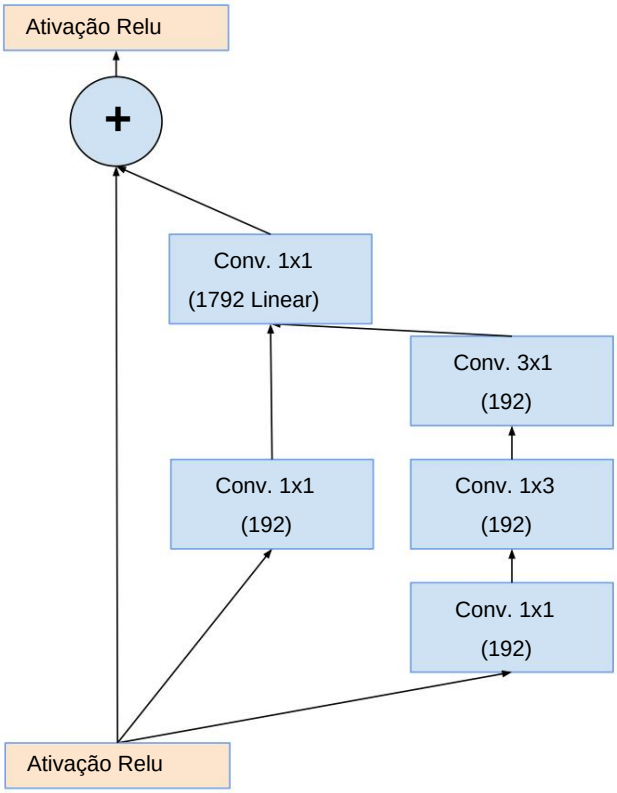

Figura 13. Esquema do módulo de grade 8×8 (Inception-ResNet-C) da rede Inception-ResNet-v1.

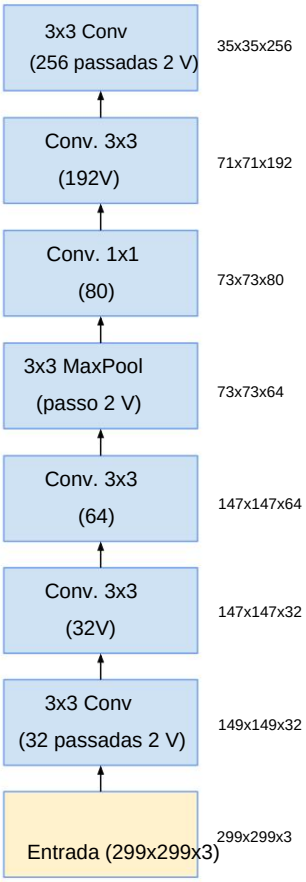

Figura 14. A haste da rede Inception-ResNet-v1.

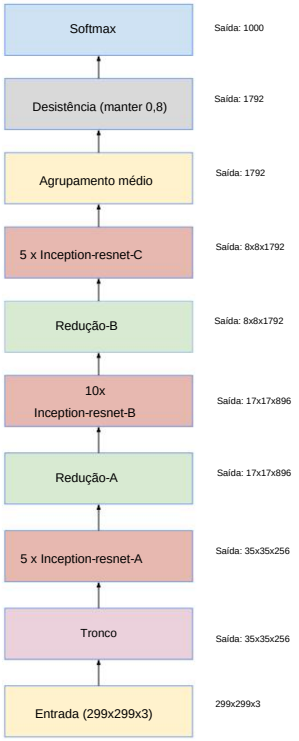

Figura 15. Esquema para redes Inception-ResNet-v1 e Inception-ResNet-v2. Este esquema se aplica a ambas as redes, mas os componentes subjacentes são diferentes. Inception-ResNet-v1 usa os blocos descritos nas Figuras 14, 10, 7, 11, 12 e 13. Inception-ResNet-v2 usa os blocos descritos nas Figuras 3, 16, 7,17, 18 e 19. A saída os tamanhos no diagrama referem-se às formas do tensor do vetor de ativação de Inception-ResNet-v1.

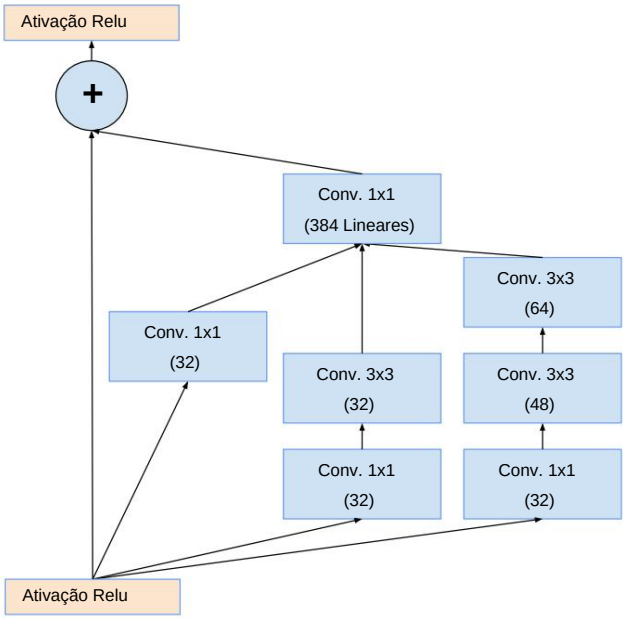

Figura 16. Esquema do módulo de grade 35 × 35 (Inception-ResNet-A) da rede Inception-ResNet-v2.

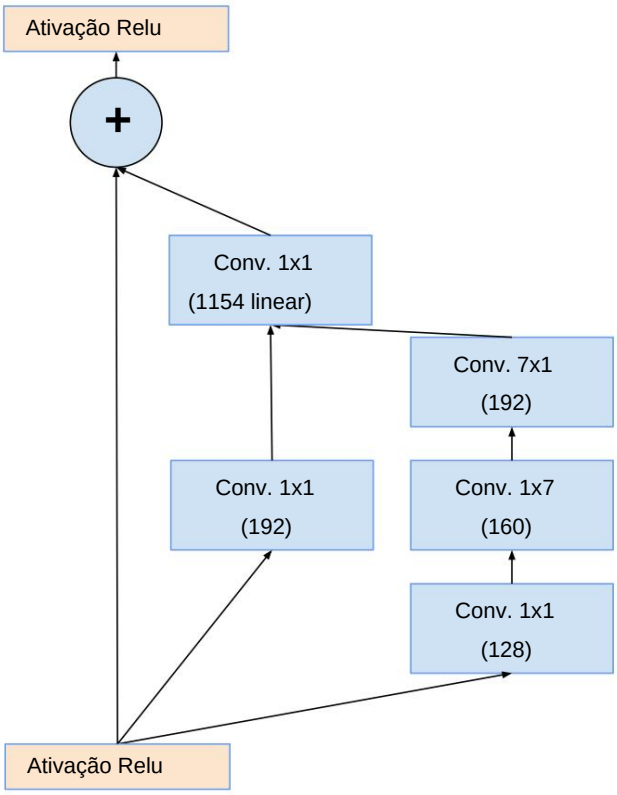

Figura 17. Esquema do módulo de grade 17 × 17 (Inception-ResNet-B) da rede Inception-ResNet-v2.

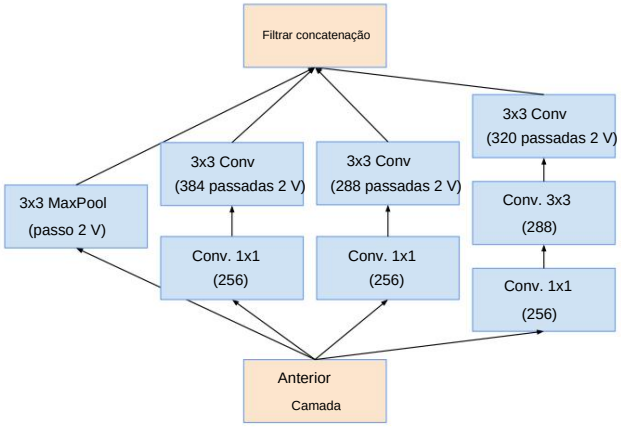

Figura 18. Esquema para módulo de redução de grade 17 × 17 a 8 × 8. Módulo Reduction-B usado pela rede Inception-ResNet-v1 mais ampla na Figura 15.

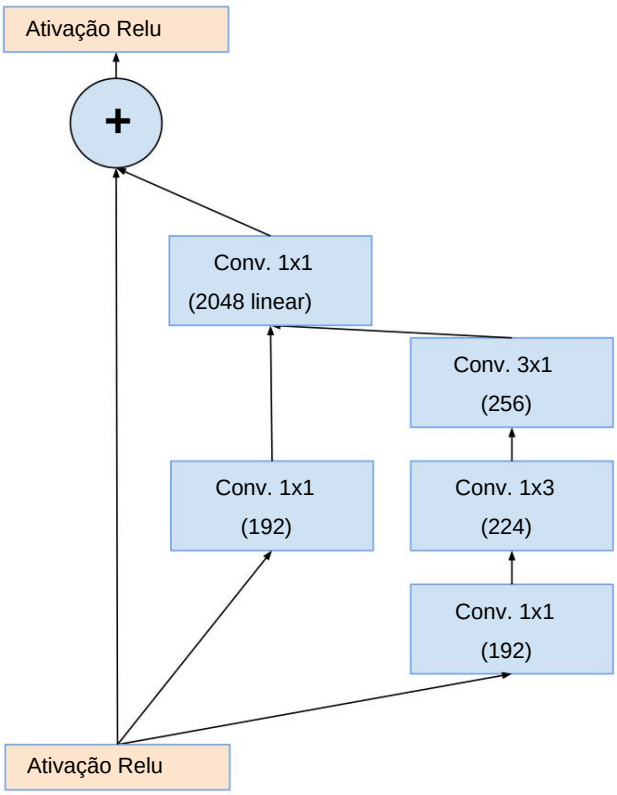

Figura 19. Esquema do módulo de grade 8×8 (Inception-ResNet-C) da rede Inception-ResNet-v2.

Rede	k	lmm		
Início-v4	192	224	256	384
Inception-ResNet-v1	192	192	256	384
Inception-ResNet-v2	256	256	384	384

Tabela 1. Número de filtros do módulo Reduction-A para o três variantes do Inception apresentadas neste artigo. Os quatro números nas colunas do artigo parametrizam as quatro convoluções da Figura 7

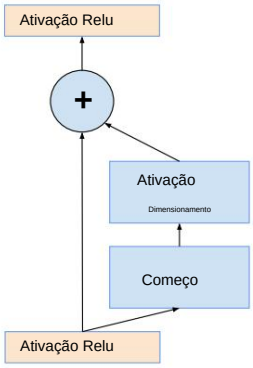

Figura 20. O esquema geral para dimensionar módulos combinados Inception-resnet. Esperamos que a mesma ideia seja útil no caso geral da resnet, onde em vez do bloco Inception é usada uma sub-rede arbitrária. O bloco de escala apenas dimensiona as últimas ativações lineares por uma constante adequada, normalmente em torno de 0,1.

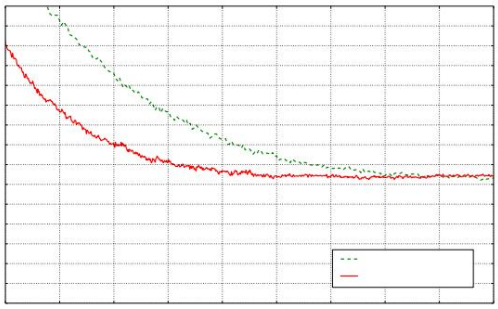

Figura 21. Evolução do erro principal durante o treinamento do Inception-v3 puro versus uma rede residual de custo computacional semelhante. A avaliação é medida em um único corte nas imagens que não estão na lista negra do conjunto de validação ILSVRC-2012. O modelo residual foi treinado muito mais rápido, mas alcançou uma precisão final um pouco pior do que o tradicional Inception-v3.

3.3. Dimensionamento dos Resíduos

Também descobrimos que se o número de filtros ultrapassasse 1000, as variantes residuais começavam a apresentar instabilidades e a rede apenas "morreu" no início do treinamento, o que significa que a última camada antes do pooling médio começou a produzir apenas zeros após um período de treinamento. algumas dezenas de milhares de iterações. Isso não poderia ser evitado, nem diminuindo a taxa de aprendizado, nem adicionando uma normalização de lote extra a esta camada.

Descobrimos que reduzir os resíduos antes de adicioná-los à ativação da camada anterior pareceu estabilizar o treinamento. Em geral, escolhemos alguns fatores de escala entre 0,1 e 0,3 para escalar os resíduos antes de serem adicionados às ativações acumuladas da camada (cf. Figura 20).

Uma instabilidade semelhante foi observada por He et al. em [5] no caso de redes residuais muito profundas e sugeriram um treinamento em duas fases onde a primeira fase de "aquecimento" é feita com taxa de aprendizagem muito baixa, seguida por uma segunda fase com alta taxa de aprendizagem. Descobrimos que se o número de filtros for muito alto, então mesmo uma taxa de aprendizagem muito baixa (0,00001) não é suficiente para lidar com as instabilidades e o treinamento com alta taxa de aprendizagem teve a chance de destruir seus efeitos. Achamos muito mais confiável apenas dimensionar os resíduos.

Mesmo quando a escala não era estritamente necessária, nunca pareceu prejudicar a precisão final, mas ajudou a estabilizar o treino.

4. Metodologia de Treinamento

Treinamos nossas redes com gradiente estocástico utilizando o sistema de aprendizado de máquina distribuído TensorFlow [1] usando 20 réplicas rodando cada uma em uma GPU NVidia Kepler. Nossos experimentos anteriores usaram momento [13] com um decaimento de 0,9, enquanto nossos melhores modelos foram alcançados usando RMSProp [16] com decaimento de 0,9 e = 1,0. Usamos uma taxa de aprendizado de 0,045, decaída a cada duas épocas usando uma taxa exponencial de 0,94. As avaliações do modelo são realizadas usando uma média contínua dos parâmetros calculados ao longo do tempo.

5. Resultados Experimentais

Primeiro, observamos a evolução dos erros de validação principais 1 e 5 das quatro variantes durante o treinamento. Após a realização do experimento, descobrimos que nossa avaliação contínua foi conduzida em um subconjunto do conjunto de validação que omitiu cerca de 1.700 entidades na lista negra devido a caixas delimitadoras inadequadas. Descobriu-se que a omissão deveria ter sido realizada apenas para o benchmark CLSLOC, mas produz números um tanto incomparáveis (mais otimistas) quando comparados com outros relatórios, incluindo alguns relatórios anteriores da nossa equipe. A diferença é de cerca de 0,3% para o erro top-1 e cerca de 0,15% para o erro top-5. Contudo, como as diferenças são consistentes, pensamos que a comparação entre as curvas é justa.

Por outro lado, executamos novamente nossos resultados de multi-corte e conjunto no conjunto de validação completo que consiste em 50.000 imagens. Além disso, o resultado final do conjunto também foi realizado no conjunto de teste e enviado ao servidor de teste ILSVRC para validação para verificar se nosso ajuste não resultou em um ajuste excessivo. Gostaríamos de salientar que esta validação final foi feita apenas uma vez e submetemos os nossos resultados apenas duas vezes no último ano: uma vez para o artigo BN-Inception e mais tarde durante a competição ILSVR-2015 CLSLOC, por isso acreditamos que o conjunto de testes os números constituem uma estimativa verdadeira das capacidades de generalização do nosso modelo.

Por fim, apresentamos algumas comparações entre várias versões do Inception e do Inception-ResNet. Os modelos Inception-v3 e Inception-v4 são redes convolucionais profundas

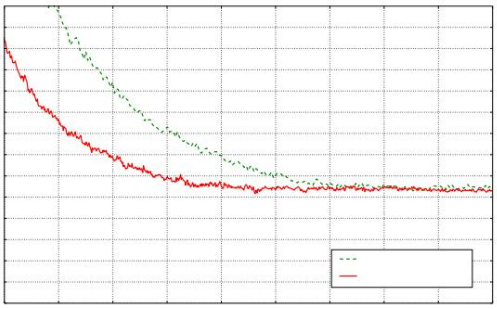

Figura 22. Evolução dos 5 principais erros durante o treinamento do Inception-v3 puro versus um Inception residual de custo computacional semelhante. A avaliação é medida em um único corte nas imagens que não estão na lista negra de o conjunto de validação ILSVRC-2012. A versão residual treinou muito mais rápido e alcançou um recall final ligeiramente melhor no conjunto de validação.

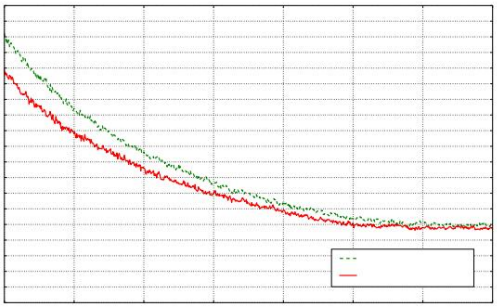

Figura 23. Evolução do erro principal durante o treinamento do Inception-v3 puro versus um Inception residual de custo computacional semelhante. A avaliação é medida em um único corte nas imagens que não estão na lista negra de o conjunto de validação ILSVRC-2012. A versão residual foi treinada muito mais rápido e alcançou uma precisão final ligeiramente melhor do que o tradicional Inception-v4.

Rede	Erro Top 1	Erro Top 5
BN-Início [6]	25,2%	7,8%
Início-v3 [15]	21,2%	5,6%
Inception-ResNet-v1	21,3%	5,5%
Início-v4	20,0%	5,0%
Inception-ResNet-v2	19,9%	4,9%

Tabela 2. Resultados experimentais de cultivo único - modelo único. Relatado no subconjunto não incluído na lista negra do conjunto de validação do ILSVRC 2012.

funciona sem utilizar conexões residuais, enquanto Inception-ResNet-v1 e Inception-ResNet-v2 são redes do estilo Inception que utilizam conexões residuais em vez de concatenação de filtro.

A Tabela 2 mostra o modelo único, colheita única top-1 e erro top-5 das várias arquiteturas no conjunto de validação.

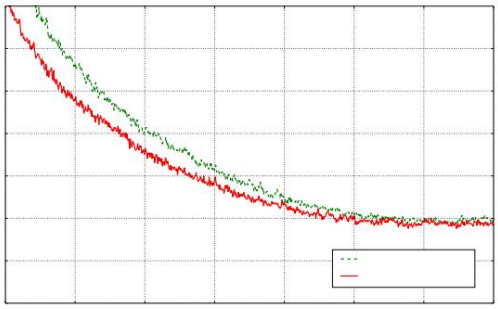

Figura 24. Evolução dos 5 principais erros durante o treinamento do Inception-v4 puro versus um Inception residual de custo computacional semelhante. A avaliação é medida em um único corte nas imagens que não estão na lista negra do conjunto de validação ILSVRC-2012. A versão residual treinada mais rápido e alcançou um recall final ligeiramente melhor no conjunto de validação.

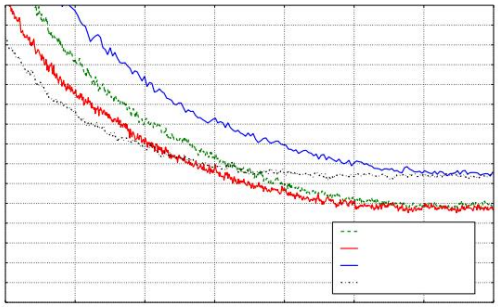

Figura 25. Evolução dos 5 principais erros de todos os quatro modelos (modelo único, colheita única). Mostrando a melhoria devido ao tamanho maior do modelo. Embora a versão residual convirja mais rapidamente, a precisão final parece depender principalmente do tamanho do modelo.

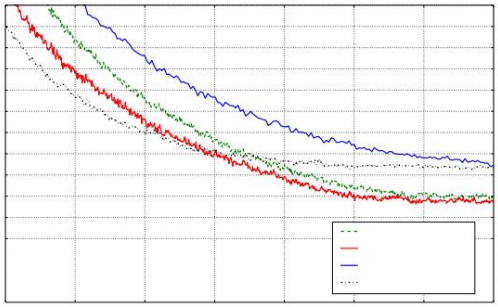

Figura 26. Evolução do erro principal de todos os quatro modelos (modelo único, colheita única). Isso mostra um quadro semelhante ao da avaliação dos 5 primeiros.

A Tabela 3 mostra o desempenho dos vários modelos com um pequeno número de colheitas: 10 colheitas para ResNet como foi relatado em [5]), para as variantes Inception, usamos o Avaliação de 12 culturas conforme descrito em [14].

Rede	Erro Top-1 de colheitas	**Erro Top-5**
ResNet-151 [5]	10	21,4% 5,7%
Início-v3 [15]	12	19,8% 4,6%
Inception-ResNet-v1	12	19,8% 4,6%
Início-v4	12	18,7% 4,2%
Inception-ResNet-v2	12	18,7% 4,1%

Tabela 3. Avaliações de culturas 10/12 - resultados experimentais de modelo único. Relatado em todas as 50.000 imagens do conjunto de validação de ILSVRC 2012.

Rede	Erro Top-1 de colheitas	**Erro Top-5**
ResNet-151 [5]	denso 19,4% 4,5%	
Início-v3 [15]	144 18,9% 4,3%	
Inception-ResNet-v1	144 18,8% 4,3%	
Início-v4	144 17,7% 3,8%	
Inception-ResNet-v2	144 17,8% 3,7%	

Tabela 4. Avaliações de 144 culturas - resultados experimentais de modelo único. Relatado em todas as 50.000 imagens do conjunto de validação do ILSVRC 2012.

Rede	**Erro Top 1** dos modelos	Erro Top 5
ResNet-151 [5]	6	– 3,6%
Início-v3 [15]	4	17,3% 3,6%
Início-v4 + 3× Inception-ResNet-v2	4	16,5% 3,1%

Tabela 5. Resultados do conjunto com 144 culturas/avaliação densa. Relatado em todas as 50.000 imagens do conjunto de validação do ILSVRC 2012. Para Inception-v4(+Residual), o conjunto consiste em um puro Inception-v4 e três modelos Inception-ResNet-v2 e foram avaliado tanto na validação quanto no conjunto de teste. O conjunto de testes o desempenho foi de 3,08% de erro entre os 5 primeiros, verificando que não nos ajustamos demais ao conjunto de validação.

A Tabela 4 mostra o desempenho do modelo único dos vários modelos utilizados. Para rede residual, o resultado da avaliação densa é relatado em [5]. Para as redes iniciais, a estratégia de 144 culturas foi usada conforme descrito em [14].

A Tabela 5 compara os resultados do conjunto. Para a rede residual pura, o resultado da avaliação densa de 6 modelos é relatado de [5]. Para as redes iniciais, 4 modelos foram montados usando a estratégia de 144 culturas conforme descrito em [14].

6. conclusões

Apresentamos três novas arquiteturas de rede em detalhe:

- Inception-ResNet-v1: uma versão híbrida do Inception que tem um custo computacional semelhante ao Inception-v3 de [15].
- Inception-ResNet-v2: uma versão híbrida mais cara do Inception com desempenho de reconhecimento significativamente melhorado mance.
- Inception-v4: uma variante pura do Inception sem resíduos conexões com aproximadamente o mesmo desempenho de reconhecimento que Inception-ResNet-v2.

Estudamos como a introdução de conexões residuais leva a uma velocidade de treinamento drasticamente melhorada para a arquitetura Inception. Também os nossos modelos mais recentes (com e sem conexões residuais) superam todas as nossas redes anteriores, apenas em virtude do aumento do tamanho do modelo.

Referências

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, GS Corrado, A. Davis, J. Dean, M. Devin, S. Ghe-mawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane,´ R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu e X. Zheng. Tensor-Flow: Aprendizado de máquina em larga escala em sistemas heterogêneos, 2015. Software disponível em tensorflow.org.

[2] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Sênior, P. Tucker, K. Yang, QV Le, et al. Redes profundas distribuídas em grande escala. Em Avanços na Informação Neural Sistemas de processamento, páginas 1223–1231, 2012.

[3] C. Dong, CC Loy, K. He e X. Tang. Aprendendo profundamente rede convolucional para super-resolução de imagens. Em Visão Computacional – ECCV 2014, páginas 184–199. Springer, 2014.

[4] R. Girshick, J. Donahue, T. Darrell e J. Malik. Ricas hierarquias de recursos para detecção precisa de objetos e semântica segmentação. Nos Anais da Conferência IEEE sobre Visão computacional e reconhecimento de padrões (CVPR), 2014.

[5] K. He, X. Zhang, S. Ren e J. Sun. Aprendizagem residual profunda para reconhecimento de imagem. pré-impressão arXiv arXiv:1512.03385, 2015.

[6] S. Ioffe e C. Szegedy. Normalização em lote: acelerando treinamento profundo da rede, reduzindo a mudança interna de covariáveis. Em Anais da 32ª Conferência Internacional sobre Aprendizado de Máquina, páginas 448–456, 2015.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, e L. Fei-Fei. Classificação de vídeo em grande escala com redes neurais convolucionais. Em Visão Computacional e Reconhecimento de Padrões (CVPR), Conferência IEEE 2014 em, páginas 1725–1732. IEEE, 2014.

[8] A. Krizhevsky, I. Sutskever e GE Hinton. Rede de imagens classificação com redes neurais convolucionais profundas. Em Avanços em sistemas de processamento de informações neurais, páginas 1097–1105, 2012.

[9] M. Lin, Q. Chen e S. Yan. Rede em rede. arXiv pré-impressão arXiv:1312.4400, 2013.

[10] J. Long, E. Shelhamer e T. Darrell. Totalmente convolucional redes para segmentação semântica. Em Anais do Conferência IEEE sobre Visão Computacional e Reconhecimento de Padrões, páginas 3431–3440, 2015.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein,

e outros. Desafio de reconhecimento visual em grande escala da Imagenet. 2014.

[12] K. Simonyan e A. Zisserman. Redes convolucionais muito profundas para reconhecimento de imagens em grande escala. Pré-impressão do arXiv arXiv:1409.1556, 2014.

[13] I. Sutskever, J. Martens, G. Dahl e G. Hinton. Sobre a importância da inicialização e do impulso no aprendizado profundo. Em Anais da 30ª Conferência Internacional sobre Aprendizado de Máquina (ICML-13), volume 28, páginas 1139–1147. Workshop JMLR e procedimentos da conferência, maio de 2013.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke e A. Rabinovich. Indo mais fundo com convoluções. Em Anais da Conferência IEEE sobre Visão Computacional e Reconhecimento de Padrões, páginas 1–9, 2015.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens e Z. Wojna. Repensando a arquitetura inicial para visão computacional. Pré-impressão do arXiv arXiv:1512.00567, 2015.

[16] T. Tieleman e G. Hinton. Divida o gradiente por uma média contínua de sua magnitude recente. COURSERA: Redes Neurais para Aprendizado de Máquina, 4, 2012. Acesso em: 05-11-2015.

[17] A. Toshev e C. Szegedy. Deeppose: estimativa de pose humana por meio de redes neurais profundas. Em Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, páginas 1653–1660. IEEE, 2014.

[18] N. Wang e D.-Y. Yeung. Aprendendo uma representação de imagem compacta e profunda para rastreamento visual. Em Avanços em Sistemas de Processamento de Informação Neural, páginas 809–817, 2013.