XXVIII Congresso {virtual} de Iniciação Científica da Unicamp 2020

Modelo para quebra de senhas por sensoriamento remoto usando Deep Learning

Bruno Eduardo Santos de Oliveira *
Josué Labaki †

***Abstract*— Nesta pesquisa foi desenvolvido um identificador de senhas digitadas em um teclado através da vibração captada com um acelerômetro próximo ao teclado utilizando técnicas de *Deep Learning*.**

A aquisição com acelerômetro foi feita com um teclado comum, com o objetivo de treinar um classificador de teclas pelo espectrograma da vibração captada. Desta forma pode-se inferir uma senha digitada pelo padrão de vibração encontrado pelo algoritmo de *Deep Learning*. Por fim, foi desenvolvido um aplicativo *Android* para captar os sinais de vibração do teclado.

Foi observado que o classificador gerado através de *Deep Learning* teve um desempenho acima do reportado na literatura, que utilizou aprendizado de máquina clássico, para teclas individuais. Entretanto ele é pouco efetivo quando o objetivo é a identificação de uma senha longa, visto que apenas um erro é suficiente para a identificação falhar.

I. Introdução

Nos *smartphones* comerciais, atualmente, existem medidas de segurança para garantir que um aplicativo só possa utilizar recursos que o usuário permita. Por exemplo, um aplicativo que tirar fotos precisa acessar a câmera do dispositivo, logo ele pede o uso ao sistema operacional e este, por sua vez, pergunta ao usuário se o aplicativo pode utilizar tal recurso, conforme a Figura 1.

Fig. 1. Janela de permissão de acesso à recurso no sistema operacional Android

Nos dois sistemas operacionais mais comuns para *smartphones*, essa não é a realidade para os sensores do aparelho. Aplicações que atuem em sistema operacional Android [1] e iOS [2] precisam simplesmente utilizar rotinas de interface com o *Hardware*, através do sistema operacional, e este garantirá que os dados sejam enviados, independente da consciência do usuário sobre qual aplicação está aquisitando dados dos sensores ou com que frequência está ocorrendo esta aquisição.

*bruno97br@gmail.com
†labaki@fem.unicamp.br

II. Objetivos

O projeto de pesquisa "Modelo para quebra de senhas por sensoriamento remoto usando *Deep Learning*", busca identificar quais teclas foram pressionadas por um teclado próximo ao dispositivo através da análise da vibração captada por um acelerômetro com propriedades similares à de um *smartphone*. Este sinal captado é processado por uma algoritmo de *Deep Learning*, de forma a detectar qual padrão de vibração identifica cada tecla pressionada no teclado.

O principal objetivo deste projeto é descobrir se é possível inferir uma senha digitada em um teclado a partir das vibrações captadas, sendo que este teclado foi previamente mapeado com um algoritmo de aprendizado de máquina.

III. Revisão bibliográfica

A primeira etapa do projeto foi a revisão do estado da arte da inferência de teclas apertadas utilizando recursos dos *smartphones*. Os principais estudos relacionados ao assunto utilizaram aprendizado de máquina, mas de pouca profundidade, portanto os dados captados do *smartphone* tiveram que passar por uma escolha de características, definidas a priori, e essas características alimentaram as redes neurais utilizadas.

Berger [3] fez a inferência, para palavras inglesas, acessando o microfone do dispositivo e analisando padrões de repetição de caracteres pressionados. Logo seu resultado foi significativamente influenciado pela quantidade de caracteres repetidos e quantidade de caracteres totais na palavra (visto que uma palavra de mais caracteres tem uma probabilidade maior de ter caracteres repetidos). Obteve mais de 90% de acurácia em todas as palavras que possuem mais de duas

repetições de caracteres ou que possuíssem mais que 12 caracteres. Em contrapartida, para palavras com 7 caracteres tiveram uma acurácia de 55% e palavra com menos caracteres não foram avaliadas. Destaca-se, neste trabalho, que a inferência não precisou de um pré-treinamento da rede e uma amostra de 20 segundos de gravação do teclado é suficiente para inferir as teclas pressionadas

Marquardt [4] analisou a qualidade da inferência de teclas pressionadas utilizando a vibração do teclado captada pelo acelerômetro do *smartphone*. Observou que a resolução do sinal captado é fortemente influenciado pela versão do dispositivo, e a frequência de amostragem máxima em seu experimento era de $100Hz$. Utilizando uma rede neural simples, obteve uma acurácia de 26% o que não foi qualificado como suficiente para inferência de informação. Então desenvolveu uma técnica de inferência por pares de caracteres pressionados, separando palavras de N caracteres em $N-1$ pares de caracteres sucessivos. Os pares identificados foram mapeados em um dicionário de palavras inglesas e, através de um sistema de pontuação, inferiu qual a palavra mais provável que corresponde aos pares de caracteres detectados. Obteve ao final um algoritmo que conseguiu retirar em média 80% das palavras digitadas.

Um algoritmo de rede neural usualmente precisa de uma etapa de pré-processamento onde apenas as característica úteis do sinal de entrada são extraídos e utilizados para o treinamento. Isto implica que a decisão de quais características de uma determinada entrada são "úteis"afeta diretamente o resultado.

Uma outra abordagem consiste em concatenar mais camadas de neurônios, de forma que a própria rede pondere quais são as características mais relevantes. Conforme a quantidade de camadas de nerônios aumenta, temos um poder maior de abstração entre as entradas e as categorias de cada entrada, essa é a proposta geral do *Deep Learning*. Como desejamos uma alta abstração (a partir da vibração do teclado, identificar qual tecla foi pressionada) utilizamos *Deep Learning* nesta pesquisa.

Para que o algoritmo seja capaz de separar o sinal distinto de cada tecla e despreze os ruídos, precisamos de um banco de dados grande o suficiente para o treinamento. Além disso, é preciso garantir que o algoritmo esteja treinado não só para as amostras utilizadas no treinamento como para amostras novas. Um algoritmo cujo erro é grande para amostras do grupo de treinamento está sub-treinado (*underfitting*) e um algoritmo que performa bem para o grupo de treinamento e mal para novas amostras está sobre-treinado (*overfitting*).

A única abstração utilizada entre o sinal e a rede neural foi utilizar os espectrogramas dos sinais ao invés do sinal bruto, desta forma fica mais explícito à rede que as teclas podem ser identificadas pelo decaimento de cada frequência no espectrograma.

Como os padrões das teclas podem estar transladados no tempo (já que cada amostra pode ter janelas de tempo com ruído antes ou depois do sinal), foi utilizado uma combinação de redes de convolução, mesma técnica utilizadas para classificar os elementos de uma imagem.

IV. Materiais e métodos

A. Dados amostrados em um teclado

1) Captação dos dados: Para a amostragem da vibração do teclado, foi utilizado um teclado mecânico da fabricante *Dell* conforme a Figura 2, um acelerômetro profissional triaxial *DeltaTron* com seu *hardware* para aquisição de dados, onde o sinal é amostrado e enviado à um software, conforme as Figuras 3 e 4.

Fig. 2. Teclado e acelerômetro na bancada de aquisição

O teclado é fixado à bancada utilizando uma fita dupla-face para que não deslize durante o experimento sem influenciar nos resultados e o acelerômetro é fixado utilizando uma cera para esta finalidade.

A captação dos sinais ocorreu apertando as teclas pausadamente, de forma a não sobrepor dois sinais, e com a mesma intensidade em que se utiliza o teclado usualmente. Como o acelerômetro é triaxial, pode-se captar as 3 dimensões da vibração e, inicialmente, as 3 seriam utilizadas para o treinamento da rede.

Dez teclas foram escolhidas de forma a mapear a parte alfabética do teclado, conforme a Figura 5. Os sinais da barra de espaços foram posteriormente descartados pois esta possui um mecanismo de três pontos de apoio, ao contrario das letras que possui apenas um. Logo a posição onde a tecla era apertada alterava muito o sinal.

O sistema de aquisição fornecido registrou os sinais em bateladas de 10 pressionamentos por vez. Então esses sinais foram separados posteriormente utilizando um limiar de energia. A energia de um sinal discreto é calculado pela Equação 1 e, quando era encontrado um trecho (janelamento de 100 milissegundos conforme Berger e Marquardt [3] [4]) com energia 3 vezes maior que a energia do ruído, o trecho era separado com uma amostra de tecla.

$$E_s = \sum_{n=-\infty}^{+\infty} |x(n)|^2 \tag{1}$$

Para diminuir os efeitos de erros acumulados foram feitas bateladas de 10 pressionamentos por cada tecla e esse processo se repetiu 7 vezes. Durante a pesquisa foi estimado que um número suficiente de amostras seria de 60 amostras por categoria a ser classificada e foram utilizadas 70 amostras por tecla como uma margem adicional.

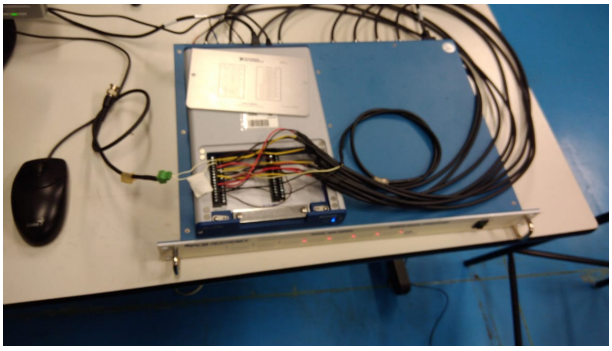

Fig. 3. *Hardware* para aquisição do sinal do acelerômetro

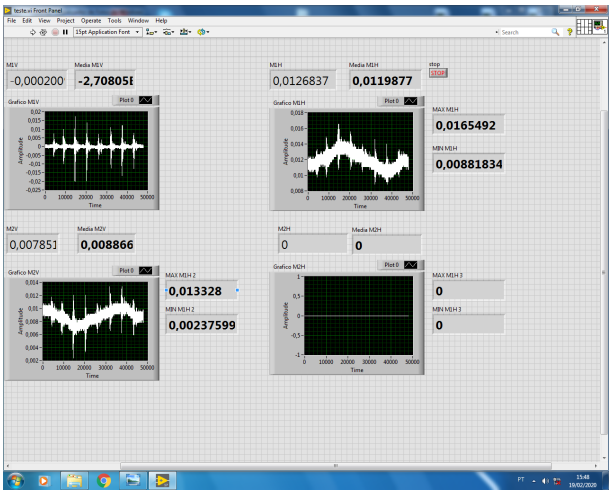

Fig. 4. *Software* para aquisição do sinal do acelerômetro

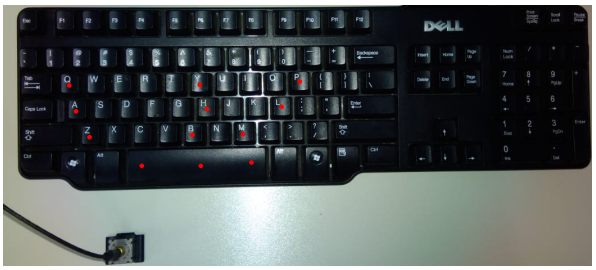

Fig. 5. Pontos vermelhos indicam onde cada (e qual) tecla foi pressionada na captação dos dados

Além da barra de espaços, as teclas escolhidas foram "q", "y", "p", "a", "g", "l", "z", "b"e "m".

2) Treinamento da rede: Inicialmente foi feito o espectrograma de cada uma das amostras, como um espectrograma é função de alguns parâmetros (tipo de janelamento e largura das janelas, principalmente) foram feitas algumas combinação com um grupo de avaliação cruzada para identificar qual seria a melhor combinação (os resultados de avaliação de rede utilizam o sub-conjunto de avaliação e, portanto, não ficam enviesados pela avaliação cruzada).

Da mesma forma a combinação de camadas de neurônios foi obtida, otimizando os resultados para um sub-conjunto distinto do grupo de treinamento e de avaliação final. A combinação com melhor resultado é apresentada na seção de Resultados Obtidos.

Após obter o banco de dados com os espectrogramas é necessário aleatorizar a ordem das amostras no banco de dados para diminuir a tendência de que cada sub-conjunto tenha mais ou menos amostras de alguma categoria o que pode causar sobre-treinamento amostras muito recorrentes e sub-treinamento nas menos recorrentes. Consequentemente, o resultado final irá variar conforme a aleatorização. Portanto os testes foram repetidos até que a média entres as acurácias de cada execução convergisse para algum valor. A proporção de dados utilizada para os sub-conjuntos foi de 70% para treinamento, 20% para validação cruzada e 10% para avaliação.

A priori espera-se que ao utilizar as 3 dimensões do acelerômetro a acurácia seria maior que utilizar apenas a dimensão principal (paralela ao movimento da tecla). Portanto foram feitos testes com as duas configurações e seus resultados foram anotados.

B. Password Cracking

Como os classificadores possuem diferentes acurácias conforme a aleatorização, foram gerados 10 classificadores e 130 senhas aleatórias com as letras utilizadas no grupo de avaliação, divididas em 13 grupos de 10 onde, em cada grupo, a senha possui n caracteres, com $1 <= n <= 13$.

Em uma iteração de *Password Cracking*, o algoritmo seleciona uma amostra do sub-conjunto de avaliação para cada caractere de uma senha e os une, conforme a Figura 6 para a senha gerada "zgqm".

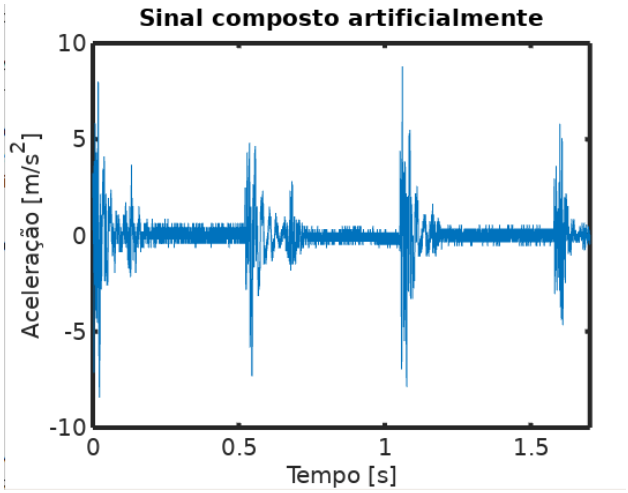

Fig. 6. Sinal composto para a senha *zgqm*

O sinal composto artificialmente então passa pelo algoritmo de avaliação, onde é separado em n caracteres e

classificados pela rede. Os resultados de sucesso são anotados juntamente da quantidade de teclas e a compilação é apresentada na seção Resultados Obtidos.

C. Aplicativo

Utilizando o *framework Expo* foi gerado um aplicativo para captar a vibração do teclado através do acelerômetro de um *smartphone*. Este framework permite gerar aplicativos para Android e iOS sem alterações no código-fonte. O aplicativo faz a aquisição na maior taxa de amostragem possível e envia os dados para um servidor *cloud*. Os sinais captados foram analisados e são apresentados na seção de Resultados.

V. Resultados Obtidos

A. Resultados da aquisição do teclado

Um exemplo do sinal captado é apresentado na Figura 7. Pode-se observar que existem dois picos dignificativos no sinal. O primeiro é causado ao pressionar a tecla, o segundo acontece quando a tecla retorna a sua posição original pelo sistemas de molas do teclado.

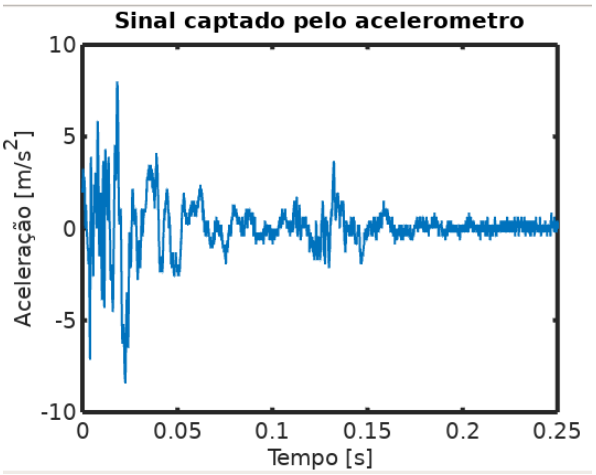

Fig. 7. Sinal captado pelo acelerômetro após pressionar uma tecla no eixo paralelo ao movimento da tecla

A relação de amplitude entre os picos é característica em cada teclada, seja por diferenças nos sistemas mecânicos, seja pela dissipação de cada sinal até que ele chegue ao acelerômetro.

Para obter o espectrograma, foi utilizado o janelamento Hann e o menor número de amostras entre colunas de *FFTs* adjacentes, obtendo uma matriz com a maior resolução possível no tempo e na frequência. Entretanto matrizes grandes são custosas para operar e serão posteriormente manipuladas por uma camada de *Pooling*. Desta forma pode-se normalizar sinais com variados tempos totais de sinal e com diferentes taxas de aquisição. Isto é necessário visto que teclas mais afastadas do acelerômetro dissipam mais rápido e para a aquisição em um *smartphone* a taxa de amostragem será menor. O espectrograma é apresentado na Figura 8

O conjunto de camadas otimizado encontrado foi:

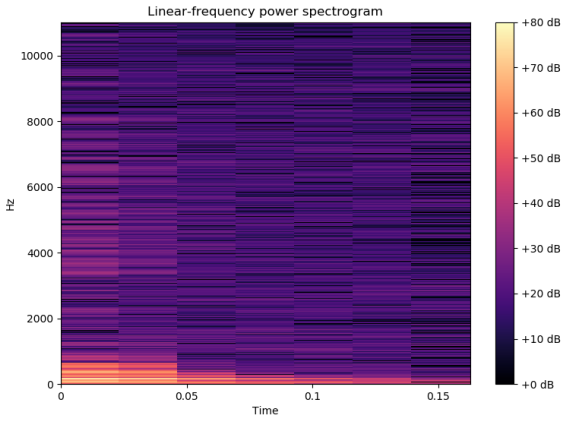

Fig. 8. Espectrograma do sinal da Figura 7

- MaxPooling2D: Janelamento em função da dimensão da entrada de forma que o tensor resultante seja de 100x100
- BatchNormalization;
- Conv2D: 48 filtros de tamanho $4x4$
- Dropout: 25%, mantendo 75% da camada anterior
- Flatten;
- Dense: ativação *softmax* e quantidade de neurônios igual à quantidade de teclas treinadas (9 neste experimento)

Para os sinais compostos das três dimensões, a acurácia média do grupo de avaliação foi de 49% com desvio padrão de 5%. Para os sinais de uma dimensão (paralela ao movimento da tecla), a acurácia foi de 70% com desvio padrão de 6%.

Este resultado mostra que, embora a utilização de um sinal tridimensional possa conter informações adicionais de ondas normais e cisalhantes, o aumento na quantidade de *features* por amostra torna mais difícil para o algoritmo convergir a um resultado ótimo.

A curva de custo média por épocas é apresentada na Figura 9. Observa-se que, embora o resultado tenha sido suficiente para classificar as teclas com uma boa acurácia, o sistema ainda está sobre-ajustado, pois a curva do grupo de validação estagnou enquanto a do grupo de treinamento ainda decaia. Como não foi encontrada nenhuma combinação de camadas que otimizasse esse resultado, pode-se supor que para reduz ainda mais o erro seria necessário um *dataset* ainda maior.

B. Password Cracking

Para o conjunto gerado de 130 senhas de 1 à 13 caracteres, foi observado que há um decaimento exponencial conforme a quantidade de caracteres na senha. A Figura 10 apresenta a acurácia das 130 combinações de senhas (de 1 à 13 caracteres) para 10 retreinamentos da rede. A melhor curva exponencial que aproxima o modelo possui coeficiente de determinação (conhecido como R^2) igual à 0.993 e equação $y = 1.01 * e^{(}-0.363 * n)$ onde y é a acurácia e n a quantidade de caracteres.

Portanto, embora a acurácia para identificar cada tecla individualmente tenha sido muito boa, ao tentar identificar

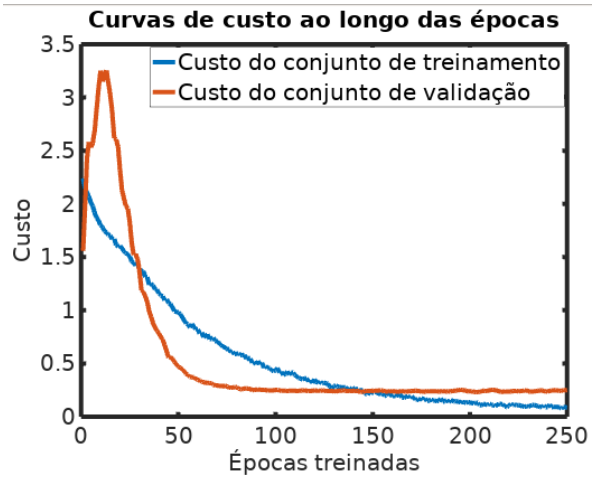

Fig. 9. Erro (ou custo) de dos sub-conjuntos de treinamento e validação em função das épocas

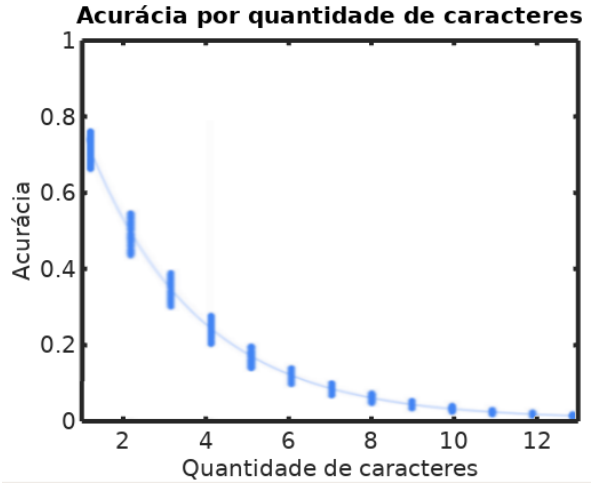

Fig. 10. Acurácia por quantidade de caracteres em uma senha gerada aleatoriamente

todas as teclas pressionadas em uma senha de 6 caracteres o algoritmo irá acertar a senha em 11.4% dos casos. Para fins de comparação, a chance de acertar uma senha de 6 caracteres (formadas pelas mesmas 9 letras do banco de dados) de forma aleatória é de 0.0002%.

C. Aplicativo

O aplicativo utilizado conseguiu obter dados com intervalos de até 10 milissegundos, ou seja, 100 Hz de aquisição, igualmente à Marquardt[4]. Desta forma pode-se assumir que os dados podem ser aquistados sem nenhum problema em relação à taxa de amostragem da teoria. A resolução do sinal também foi suficiente próxima aos dados do acelerômetro profissional, pois este ultimo foi truncado para 7 casas decimais para salvar e utilizar os dados como *float* ao invés de *double* para reduzir o consumo de memória.

VI. Conclusão

Os dados obtidos com o teclado e um acelerômetro profissional tiveram uma boa acurácia para teclas isoladas. Entretanto para identificar uma senha completamente é necessário identificar todos os caracteres e, para uma senha de 6 caracteres, por exemplo, a probabilidade do classificador acertar todas as letras é de 11,4%. Esses resultado também pode passar por um pós-filtro semântico, o que poderia ajudar a identificar senhas mais humanas e menos aleatórias. Por exemplo, a palavra *blablablq* poderia ser transformada em *blablabla*.

Adicionalmente foi observado que, embora as ondas de pressão normal e cisalhante podem se propagar em velocidades diferentes, a inclusão de mais eixos na aquisição de sinais não melhorou os resultados. Isto acontece pois cada amostra triplicará a quantidade de *features* e aumenta a dificuldade do algoritmo convergir a uma resposta ótima para a mesma quantidade de amostras.

A continuação deste trabalho pode incluir outras características humanas ao teclar, como o tempo entre teclas próximas e distantes, além de aumentar o banco de dados de amostras.

Referências

[1] I. Google, "Sensors overview -|- android developers." `https://developer.android.com/guide/topics/sensors/sensors_overview`, 2019. Accessed: 2019-12-01.

[2] I. Apple, "Getting raw accelerometer events | apple developer documentation." `https://developer.apple.com/documentation/coremotion/getting_raw_accelerometer_events`, 2019. Accessed: 2019-12-01.

[3] Y. Berger, A. Wool, and A. Yeredor, "Dictionary attacks using keyboard acoustic emanations," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 245–254, ACM, 2006.

[4] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM conference on Computer and communications security*, pp. 551–562, ACM, 2011.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[6] M. Kaya, "An algorithm for image clustering and compression," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 13, no. 1, pp. 79–92, 2005.

[7] G. Grigg, A. Taylor, H. Mc Callum, and G. Watson, "Monitoring frog communities: an application of machine learning," in *Proceedings of eighth innovative applications of artificial intelligence conference, Portland Oregon*, pp. 1564–1569, 1996.

[8] G. Madureira and A. E. Ruano, "A neural network seismic detector," *Acta Technica Jaurinensis*, vol. 2, no. 2, pp. 159–170, 2009.

[9] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 5, pp. V–429, IEEE, 2003.

[10] C. N. Silla Jr, A. L. Koerich, and C. A. Kaestner, "The latin music database.," in *ISMIR*, pp. 451–456, 2008.

[11] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[12] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proceedings of the 39th annual meeting on association for computational linguistics*, pp. 26–33, Association for Computational Linguistics, 2001.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.