Akustični napad bočnog kanala nad tastaturama zasnovan na dubokom učenju

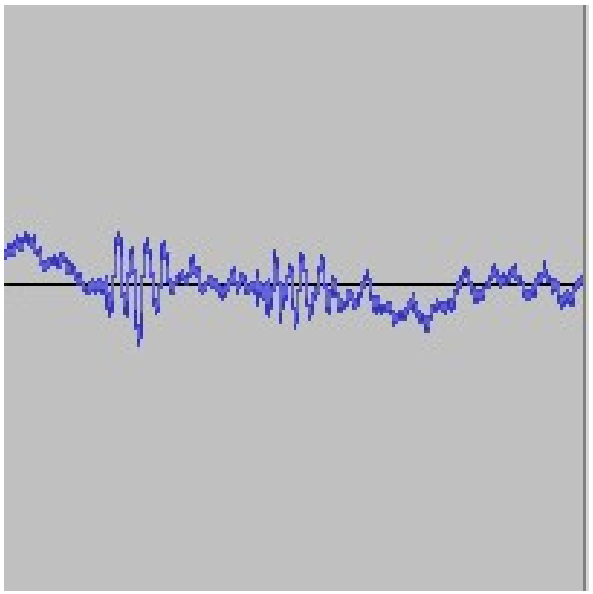
Momir Milutinović SV 39/2021

5. jul 2024.

Sadržaj

Problem

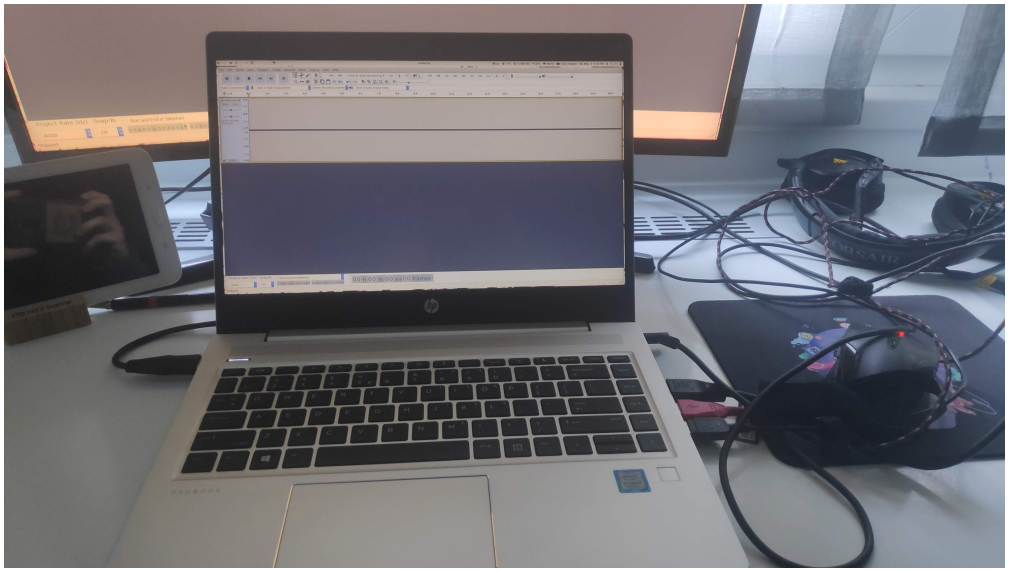
- Odrediti koji taster na tastaturi je pritsinut na osnovu zvuka pritiska tastera



Slika: Talasni oblik zvuka pritsika tastera

Prikupljanje podataka

- Svi tasteri za slova engleske su pritisnuti po 50 puta, različitim jačinama i menjajući prst kojim se pritiska taster
- Snimci su snimljeni ugrađenim mikrofonom laptopa
- Strani zvuci su dospeli u neke snimke



Slika: Laptop koji je korišćen za snimanje

Izdvajanje pritisaka tastera

- Postupak opisan u radu „A Practical Deep Learning-Based Acoustic Side Channel Attack on Keyboards“

1. Izračunava se energija signala primenom brze Furijeve transformacije i sumiranjem dobijenih koeficijenata
2. Definiše se minimalna vrednost energije, koja ukazuje na prisustvo pritiska tastera
3. Za svaki momenat u snimku gde je energija premašila prethodno defnisani prag se uzima isečak dužine 0.33 s koji počinje 0.1 s pre tog momenta
    - Izolovani pritisci tastera se ne poklapaju
4. Minimalna vrednost energije, koja ukazuje na prisustvo pritiska tastera, se postepeno menja dok se ne izdvoji tačan broj pritisaka tastera

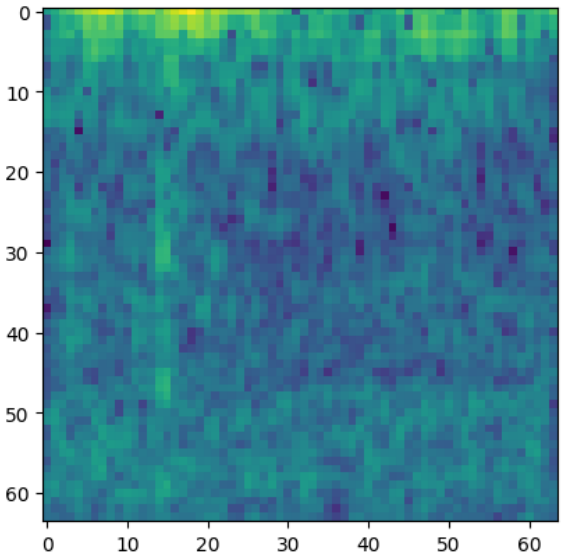
Algoritam za izdvajanje pritisaka tastera

```
def nadji_prag(snimak, pocetni_prag, korak, trazeni_broj_pritisaka_tastera):
    trenutni_prag = pocetni_prag
    pritisci_tastera = izdvoj_pritiske_tastera(snimak, pocetni_prag)
    while len(pritisci_tastera) != trazeni_broj_pritisaka_tastera:
        if len(pritisci_tastera) > trazeni_broj_pritisaka_tastera:
            trenutni_prag += korak
        else:
            trenutni_prag -= korak
        korak = korak * 0.99
        pritisci_tastera = izdvoj_pritiske_tastera(snimak, trenutni_prag)

    return (pritisci_tastera, trenutni_prag)
```

Izdvajanje osobina

- Osobine snimaka se izdvajaju pravljenjem mel spektrograma
- x-osa predstavlja vreme, y-osa frekvenciju, a boja predstavlja amplitudu određene frekvencije u datom trenutku
- Korišćena je dužina prozora 1024 i 64 mel opsega
- Slike dimenzija 64x64 piksela

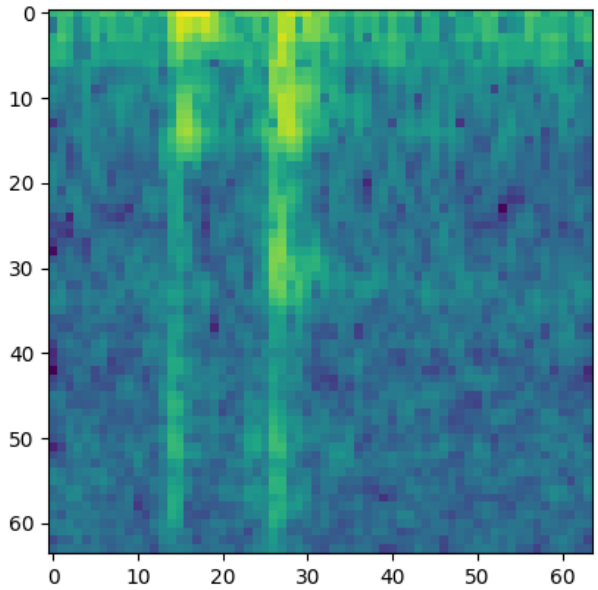


Slika: Mel spektrogram pritiska tastera

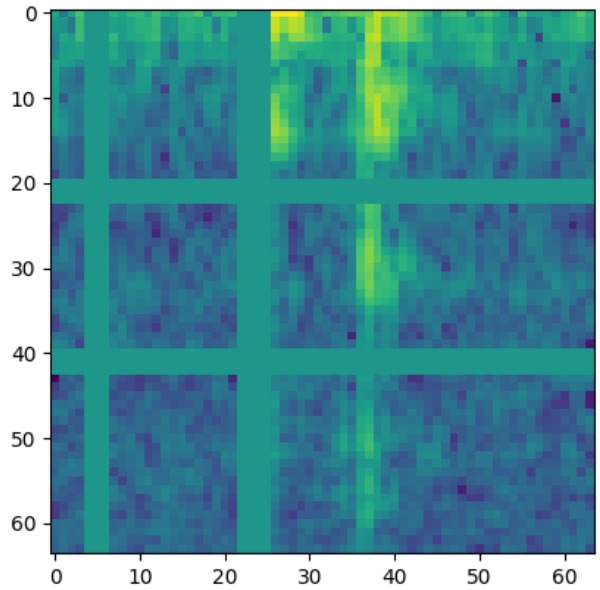
Augmentacija podataka

- Nad spektrogramima u skupu za obuku je primenjena tehnika za augmentaciju podataka nazvana SpecAugment
  1. Spektrogram se vremenski pomera za nasumični pomeraj do 30% dužine spektrograma
  2. Maskiraju se 2 nasumična odsečka ose za vreme nasumičnih širina, koje mogu biti najviše 7 piksela (malo više od 10% širine spektrograma)
     - Maskirane vrednosti se zamenjuju prosečnom vrednošću spektrograma
  3. Isto maskiranje se primenjuje i na osu za frekvenciju
- Augmentacija podataka za obuku se ponavlja u svakoj epohi

Primer spektrograma pre i nakon augmentacije podataka



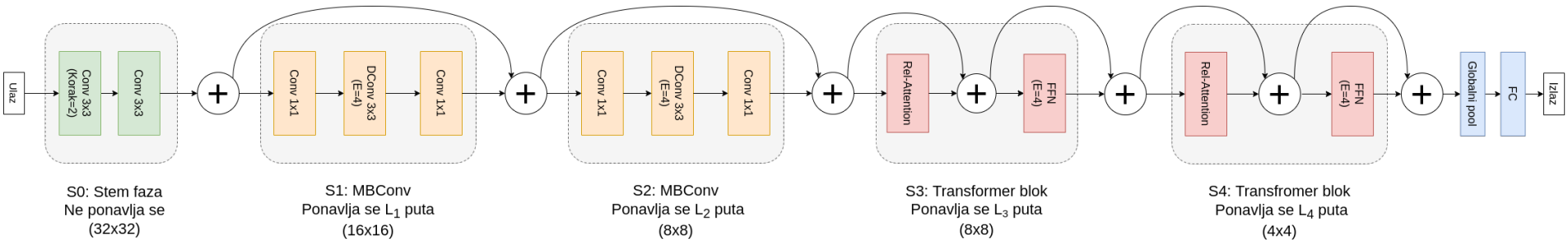
Slika: Spektrogram pre primene SpecAugment



Slika: Spetkrogram nakon primene SpecAugment

Arhitektura neurnoske mreže

- Za arhitetkturu neuronske mreže za klasifikaciju spetkrograma je izabrana CoAtNet
- CoAtNet arhitektura je implmentirana u *PyTorch* radnom okviru



Slika: Dijagram CoAtNet arhitekture (Po uzoru na dijagram iz rada „CoAtNet: Marrying Convolution and Attention for All Data Sizes“

Podela podataka

- Podaci su na podeljeni na skupove za obučavanje, validaciju i testiranje
- Odnos veličina skupova za treniranje, validaciju i testiranje je 60/20/20
- Podela je vršena na slučajan način

Izbor hiperparametara

- Razmatrane su CoAtNet-0 i CoAtNet-1 varijante CoAtNet arhitekture
- FC i konvolutivni slojevi su inicijalizovani Kaiming inicijalizacijom

Hiperparametar	Vrednost
Broj epoha	1100
Veličina podskupa	16
Funkcija greške	*Cross entropy*
Optimizacioni algoritam	*AdamW*
Maksimalni korak učenja	$5 \cdot 10^{-4}$
Minimalni korak učenja	10^{-6}
Raspored koraka učenja	Linearan
Koeficijent smanjenja težina (eng. *weight decay*)	0.1

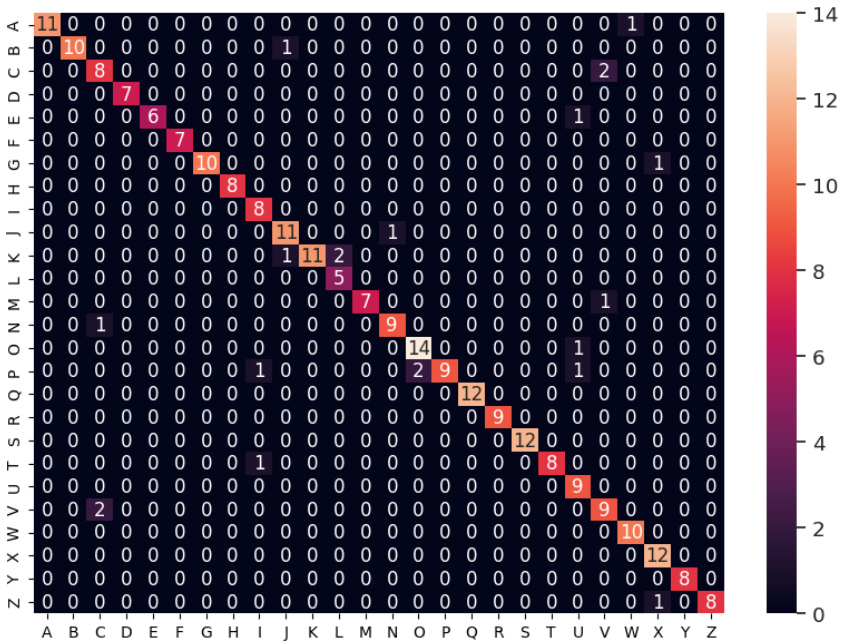
Tabela: Odabrane vrednosti hiperparametara

Rezultati

- I CoAtNet-0 i CoAtNet-1 su postigle istu maksimalnu tačnost od 94.5% na validacionom skupu
- Izlažu se rezultati za CoAtNet-0 mrežu jer je ona manja
- Nad skupom za testiranje je dobijena tačnost od 91.8%

Diskusija

- Iz analize matrice konfuzije je primećeno da se netačno predviđeni tasteri uglavnom nalaze blizu ili pored tačnih tastera
- 12 pogrešnih predikcija su bile za 1 taster udaljene od tačnog tastera, od čega je njih 10 bilo levo ili desno



Slika: Matrica konfuzije nad skupom za testiranje

Diskusija

- Takođe možemo primetiti da je došlo do preprilagođavanja.
- Harrison et al. su istrenirali CoAtNet model koji postiže 95% tačnosti nad snimcima sa telefona u blizini tastature
- Prethodno spomenuti istraživači su istrenirali model i nad snimcima iz Zoom aplikacije sa najmanjim podešavnjima smanjenja šuma i dobili su 93% tačnosti
- Akinbi et al. su napravili ConvMixer model koji ima tačnost od 92.4% nad snimcima sa telefona
- Moji rezultati ne predstavljaju poboljšanje nad postojećim, ali greške ispoljavaju iste obrasce koji su uočili Harrison et al.

Zaključak

- Obučena je efiksana mreža za klasifikaciju zvukova tastera na tastaturi
- Duboko učenje se pokazalo kao efikasna metoda za izvršavanje napada bočnog kanala nad tastaturama
- Rezultati Harrison et al. su uspešno reprodukovani

Literatura

1. Harrison, J., Toreini, E., Mehrnezhad, M. (2023, July). A practical deep learning-based acoustic side channel attack on keyboards. In 2023 IEEE European Symposium on Security and Privacy Workshops (EuroSPW) (pp. 270-280). IEEE.
2. Dai, Z., Liu, H., Le, Q. V., Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. Advances in neural information processing systems, 34, 3965-3977.
3. Taheritajar, A., Harris, Z. M., Rahaeimehr, R. (2023). A Survey on Acoustic Side Channel Attacks on Keyboards. arXiv preprint arXiv:2309.11012.
4. Akinbi, A., Deniz, E., Ismael, A. M., Rashid, Z. N., Sengur, A. (2023). Password-sniffing acoustic keylogger using machine learning. Available at SSRN 4431909.

Literatura

5. https://www.kaggle.com/code/anastasiialobanova/my-coatnet
6. https://github.com/xmu-xiaoma666/External-Attention-pytorch/blob/master/model/attention/CoAtNet.py
7. https://m0nads.wordpress.com/tag/self-attention/
8. https://github.com/chinhsuanwu/coatnet-pytorch/blob/master/coatnet.py