

# **ԵՐԵՎԱՆԻ ՊԵՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ**

**ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԿԻՐԱՌԱԿԱՆ ՄԱԹԵՄԱՏԻԿԱՅԻ ՖԱԿՈՒԼՏԵՏ  
ՊԻՍԿՐԵՏ ՄԱԹԵՄԱՏԻԿԱՅԻ ԵՎ ՏԵՍԱԿԱՆ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԱՄԲԻՈՆ**

## **Կուրսային աշխատանք**

**Թեմա՝ Նեյրոնային ցանցերում տվյալների ձևափոխությունների  
մեթոդներ**

**Ուսանող՝ Խաժակ Գալստյան**

**Ղեկավար՝ Հրանտ Խաչատրյան**

**ԵՐԵՎԱՆ 2019**

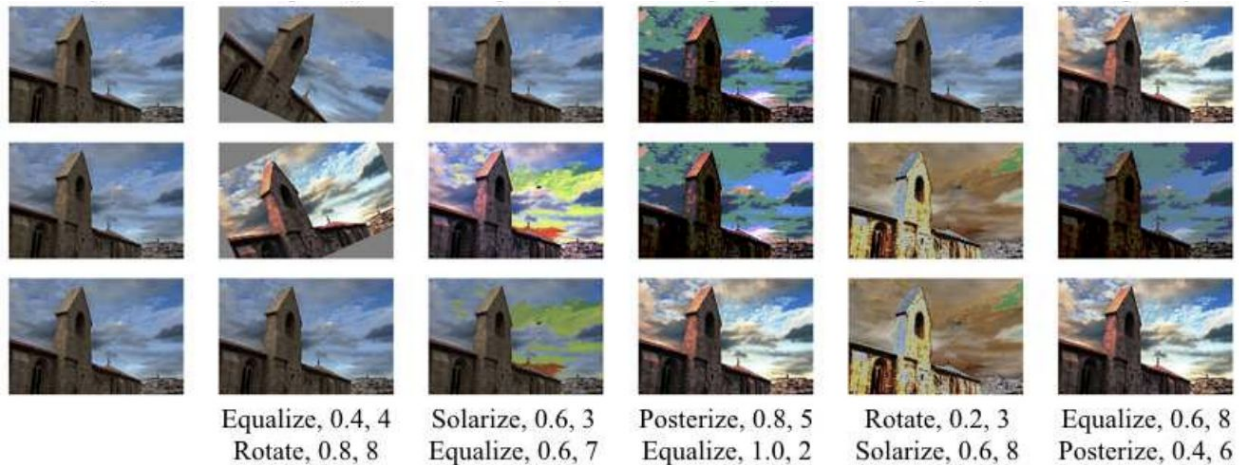
# 1. Ներածություն

Ներդրումային ցանցերի արդյունավետության մեջ շատ մեծ դեր է խաղում տվյալների քանակը և զանազանությունը: Տվյալները շատացնելու և բազմազան դարձնելու մեթոդներից է՝ տրված տվյալները ենթարկել ինչ-որ ձևափոխությունների և դիտարկել որպես նոր տվյալ, ենթադրելով, որ անհրաժեշտ առանձնահատկությունները (օրինակ՝ պիտակը) պահպանվել են: Այս աշխատանքում կդիտարկենք տվյալները (մեր դեպքում նկարները), ձևափոխությունների ենթարկելու մի քանի մեթոդներ և նրանց տված արդյունքները CIFAR-10 տվյալների բազայի վրա:

## 2. Ընդհանուր հասկացություններ

Մեր ներկայացրած մեթոդները օգտագործում են ձևափոխությունների 13 տեսակ.

- Identity: Թողնում է նկարը նույնը
- ShearX: Թեքում է նկարը  $x$  առանցքի նկատմամբ
- ShearY: Թեքում է նկարը  $y$  առանցքի նկատմամբ
- TranslateX: Նկարը տանում է աջ կամ ձախ ըստ  $x$  առանցքի
- TranslateY: Նկարը տանում է աջ կամ ձախ ըստ  $y$  առանցքի
- Rotate: Պտտում է նկարը
- Equalize: Հավասարեցնում է նկարը ըստ պիքսելների
- AutoContrast: - Նորմալիզացնում է նկարը, ամենափոքր պիքսելը դարձնում է սև, ամենամեծը՝ սպիտակ
- Color: Շատացնում կամ պակասեցնում է գույները
- Posterize: - Փոխում է գունային զամման
- Solarize: - Շրջում է ֆիքսված թվից մեծ արժեք ունեցող պիքսելների արժեքները
- Contrast: - Փոխում է պիքսելների արժեքները, հարաբերությունը թողնելով նույնը
- Brightness: - Բարձրացնում կամ իջեցնում է նկարի պայծառությունը
- Sharpness: - Փոխում է նկարի սրությունը



ձևափոխությունների օրինակներ

Մեր հիմնական խնդիրը ամեն նկարի համար օպտիմալ ձևափոխությունների և դրանց հերթականության ընտրությունն է:

### 3. Ավտոմատ Ձևափոխումներ (AA)

Google ընկերության հետազոտողների առաջարկած Auto augmentation մեթոդը [1] ընտրում է օպտիմալ ձևափոխությունները, նրանց ամեն նկարի համար ընտրվելու հավանականությունները և հերթականությունները օգտագործելով Reinforcement Learning (ամրապնդմամբ ուսուցում): Տարբերակներից լավագույնը ընտրելու համար վերցվում է այն մեկը, որի դեպքում ցանցի տված սխալանքը վերջնական տվյալների բազայի վրա նվազագույնն է: Այս մեթոդը բավականին դանդաղ է, քանի որ ընտրության միջակայքը մեծ է:

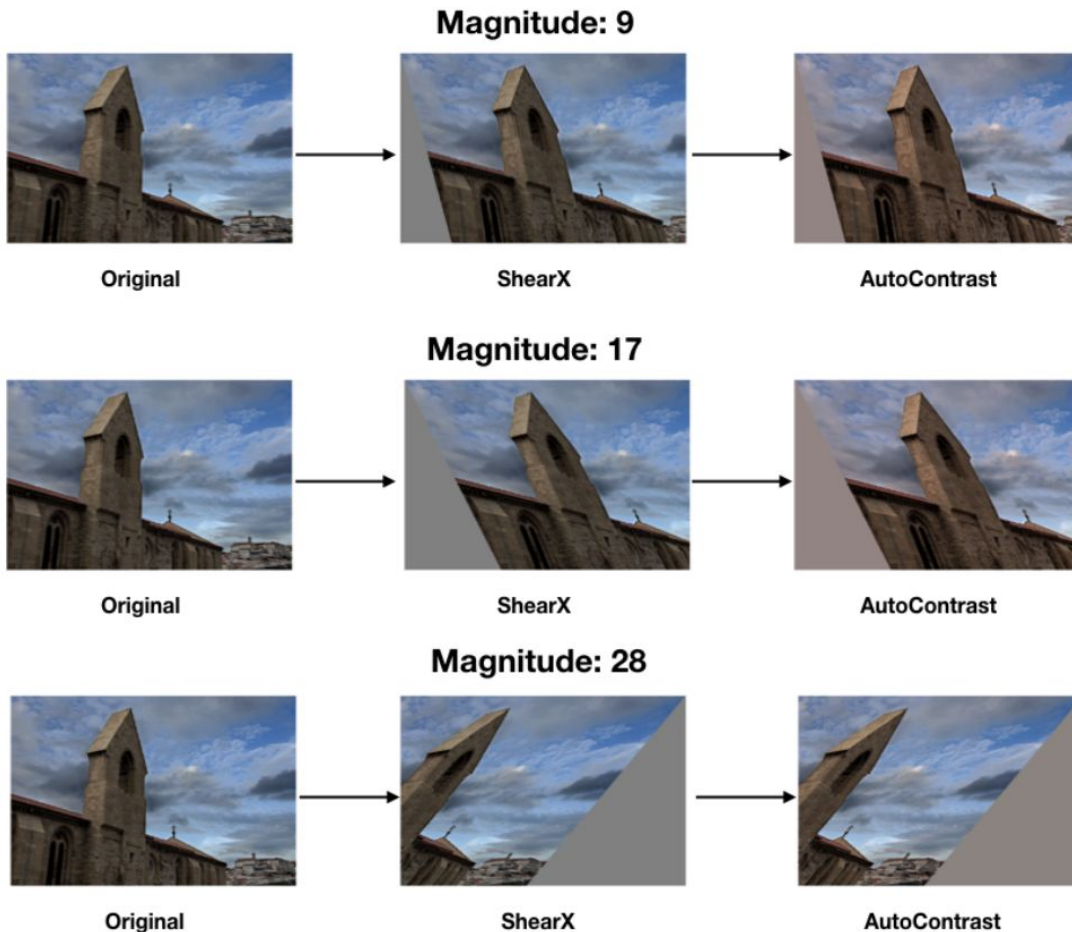
### 4. Ձևափոխություններ բնակչության հիման վրա (PBA)

Population Based Augmentation մեթոդը [2] առաջարկում է ոչ թե ֆիքսել ինչ-որ օպտիմալ ձևափոխություններ, այլ կազմել կազմել ձևափոխությունների “ժամանակացույց”: PBA-ն որոշ բազաների վրա դրսևորում է AA-ից ոչ պակաս արդյունավետություն, սակայն ավելի արագ է աշխատում:

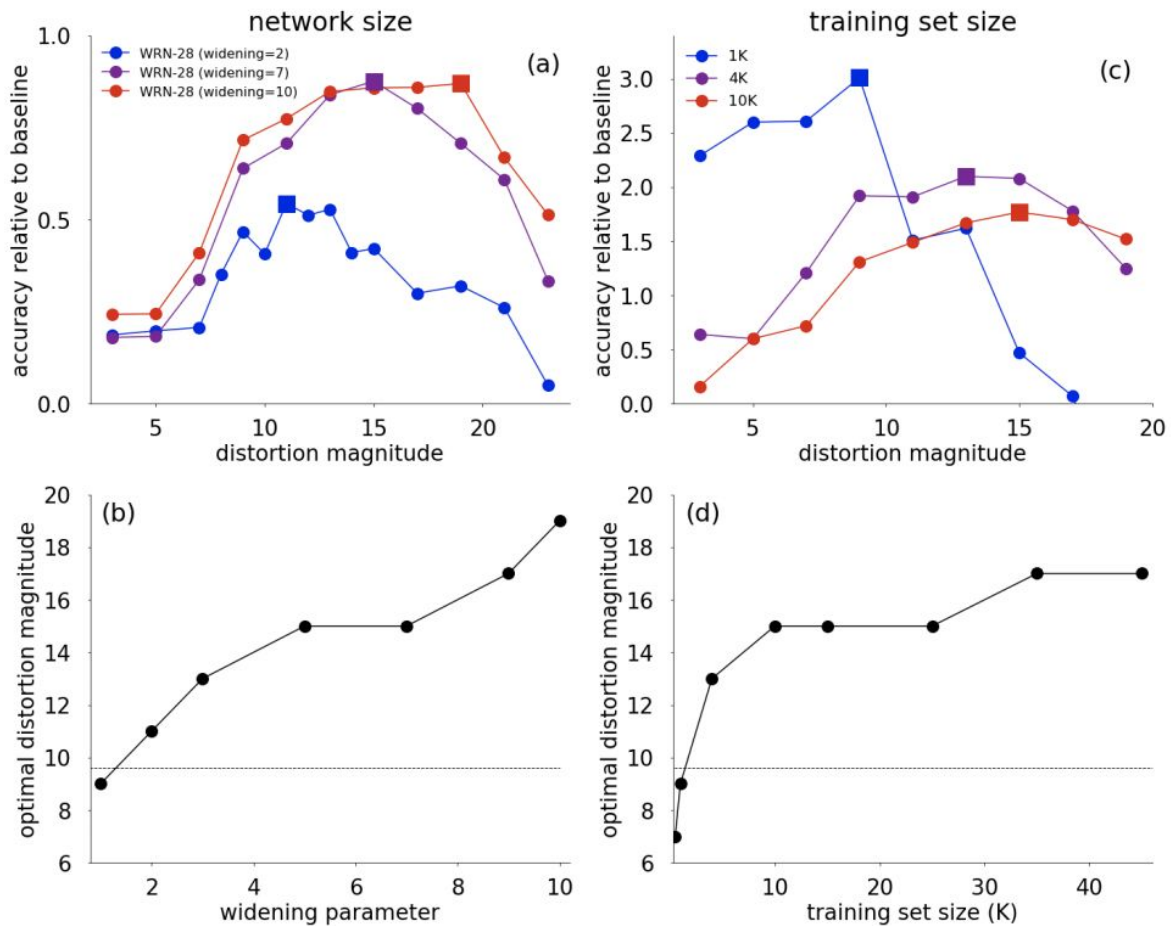
### 5. Պատահական Ձևափոխումներ (RA)

AutoAugment-ի հեղինակների առաջարկած RandAugment մեթոդը [3] առաջարկում է ընտրել օպտիմալ ձևափոխությունները համեմատաբար փոքր բազայի վրա և ընտրել մեկ

նկարի վրա կատարվող ձևափոխությունների քանակ և ուժգնություն (magnitude): Մերօրը առաջարկում է օպտիմալացնել 2 հիպերպարամետրերը՝ հաջորդաբար օգտագործվող ձևափոխությունների քանակը և ուժգնությունը:



Ուսումնասիրությունները ցույց են տվել, որ ուժգնության մակարդակի օպտիմալ արժեքը կախված է նկարից, այսինքն կարելի է օպտիմալ մակարդակը գտնել սկզբից և կիրառել նույնը բոլոր նկարների վրա առանց ձևափոխելու: Հեղինակները նաև նկատել են, որ հիպերպարամետրերի ընտրության համար բավական է դիտարկել ընդամենը 20-30 նկարից բաղկացած բազայի վրա տված արդյունքները: CIFAR-10 բազայի համար հեղինակները առաջարկում են ընտրել 3 որպես հաջորդաբար կիրառվող գործողությունների քանակ և 4 (առավելագույնը 20 է) որպես ուժգնություն: Ձևափոխությունները ունեն ընտրվելու հավասար հավանականություն ( $\frac{1}{13}$ ): Առաջարկված ալգորիթմը անհամեմատ ավելի արագ է և իրագործելի է ամենասովորական համակարգչով: Այն նաև ավելի արդյունավետ է, քան նախորդ դիտարկվածները:



## 6. Արդյունքներ

Մեթոդ	AA	PBA	RA (հոդված)	Մերը առանց ոչինչ	RA (մեր)
Ցուցադրած արդյունք	98.5	98.5	98.5	54	51

Հոդվածներում գրանցած արդյունքները օգտագործում են բարդ նեյրոնային ցանցեր: Մենք օգտագործել ենք PyTorch tutorial-ի [5] առաջարկած ցանցը ընդամենը 3 (epoch)-ով, որպեսզի հնարավոր լինի հաշվարկները կատարել սովորական համակարգչով: (Epoch)-ների սակավությունը կարող է բացատրել RA-ի ցուցադրած վատ արդյունքը մեր ցանցի վրա: Կարևոր է նաև նշել, որ RA-ը ցուցադրում է state-of-the-art արդյունք հայտնի մի քանի բազաների վրա՝ CIFAR-10/100, SVHN, ImageNet, COCO:

Մշակված ծրագիրը հասանելի է github-ում. <https://github.com/Khazhak/courseWork>

## Գրականություն

1. Cubuk, Ekin D., Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. "AutoAugment: Learning Augmentation Strategies From Data." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 113-123. 2019.
2. Ho, Daniel, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. "Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules." arXiv preprint arXiv:1905.05393 (2019).
3. Cubuk, Ekin D., Barret Zoph, Jonathon Shlens, and Quoc V. Le. "RandAugment: Practical automated data augmentation with a reduced search space." arXiv preprint arXiv:1909.13719 (2019).
4. Ildoo Kim, Unofficial PyTorch Reimplementation of RandAugment, <https://github.com/ildoonet/pytorch-randaugment>
5. PyTorch tutorials, [https://github.com/pytorch/tutorials/blob/master/beginner\\_source/blitz/cifar10\\_tutorial.py](https://github.com/pytorch/tutorials/blob/master/beginner_source/blitz/cifar10_tutorial.py)