Spring 2022

Creation of a Neural Network for the American Sign Language to Russian Translation App

John T. Simmons

Creation of a Neural Network for the American Sign Language to Russian Translation App

John Simmons
Bachelor of Arts in Multidisciplinary Studies
with focus in Machine Learning, Artificial Intelligence, and Russian Language

Outline

I. Concept
II. Timeline
III. Selected Terminology
IV. Outcomes and Demonstration
V. Technology
VI. Comparison with Similar Research Project
VII. Language Data Selection
VIII. Language Skills
IX. Challenges
X. Future Directions

I. Concept

1. A large population of people utilize American Sign Language for their primary method of communication.
2. No commercially available product is available for these people for when they need to communicate with speakers of a foreign language.
3. We must investigate methods to make communication between these two parties easier and more accessible.
4. By using a neural network to classify images of American Sign Language letters, we can build a service to make translation of American Sign Language into foreign languages possible.

II. Project Timeline

1. Spring 2021 - Identify Advisors, select prospective topic, gather initial resources
2. Summer/Fall 2021 - Narrowed down the scope of the project, identify and map out tasks, gather references and gathered resources, obtained technology.
3. Fall/Winter 2021 - Initial testing of concepts and technological applications.
4. Winter/Spring 2022 - Refining concepts, adjusting AI parameters, cyclical evaluation of AI output, conceptualizing ways to visualize the data. Delivery technologies investigated.
5. April 2022 - Concept, Demonstration, Proof of Concept.

III. Selected Terminology

1. **Epoch:** A measure of the number of times that the full set of training data is used to train our neural network. For example, three epochs means that the entire dataset was used three times to train the neural network.
2. **Loss:** A measure of how "incorrect" our neural network was in our last training batch. Less loss means our network is more accurate, and allows us to make sure our training is improving model accuracy.
3. **Tensor:** A form of data that can be interpreted by our neural network. A numerical value between negative infinity and positive infinity.

IV. Outcomes and Demonstration

johntsimmons.github.io/znak-frontend

```python
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(1, 12, 5, padding = 2)
        self.bn1 = nn.BatchNorm2d(12)
        self.pool = nn.MaxPool2d(2,2)
        self.conv2 = nn.Conv2d(12, 32, 5, padding = 2)
        self.bn2 = nn.BatchNorm2d(32)
        self.conv3 = nn.Conv2d(32, 6, 5, padding = 2)
        self.bn3 = nn.BatchNorm2d(6)
        self.fc1 = nn.Linear(864,256)
        self.fc2 = nn.Linear(256, 80)
        self.fc3 = nn.Linear(80,26) #was 28 for old_model.model
        self.drop = nn.Dropout2d(0.25)

    def forward(self, x):
        x = self.pool(F.relu(self.bn1(self.conv1(x))))
        x = self.pool(F.relu(self.bn2(self.conv2(x))))
        x = self.drop(x)
        x = self.pool(F.relu(self.bn3(self.conv3(x))))
        x = torch.flatten(x, 1)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

V. Technology

1. Utilized Python programming language and PyTorch Neural Network library.
2. For front end utilized website that communicates with neural network through an API.
3. For proof of concept used Heroku to host our back end, and Github pages for our front end.
4. For fast iteration I created several tools for creating new datasets, models, and evaluating model performance.

1. Sign - 0 and Sign - 1

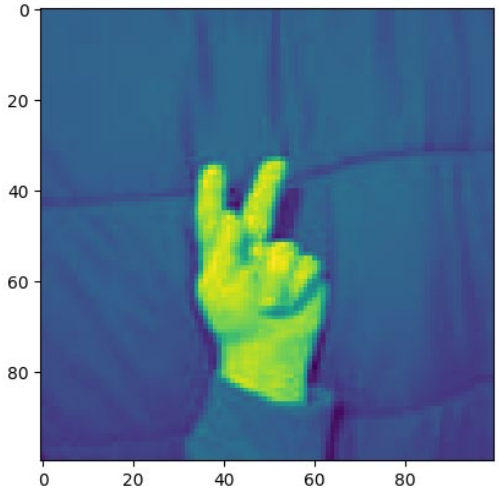
Sign - 0 (Знак-0)

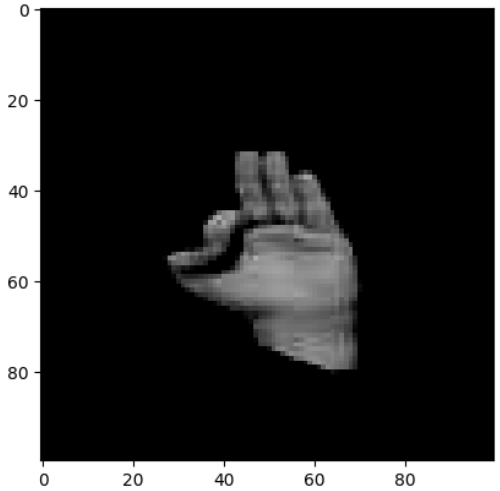
1. Linear neural network.
2. Collection of models used for research and development of candidate model.
3. No feature extraction or normalization.
4. Slow (10-20 second query time)
5. 480,000 input tensors, 28 output tensors.
    a. (400 * 400 * 3)

Sign - 1 (Знак-1)

1. Convolutional neural network.
2. Candidate model selected using grid search to optimize model.
3. Hand is normalized and “extracted” out of image background.
4. Fast (100ms-150ms query time)
5. 10,000 input tensors, 26 output tensors.
    a. (100 * 100 * 1)

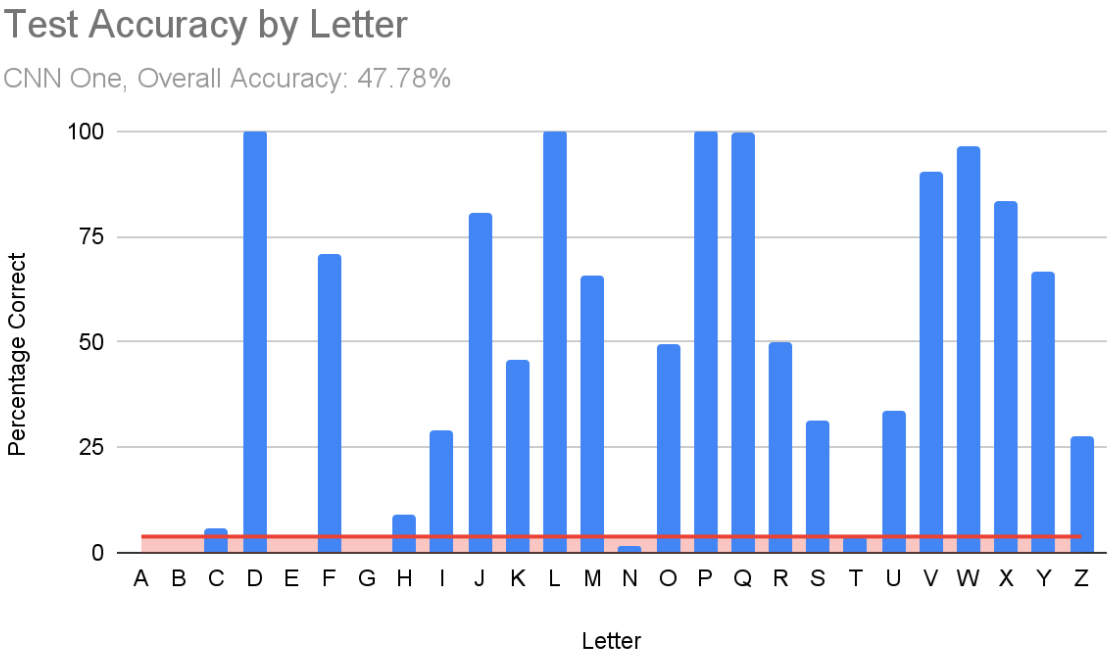
2. Select Example Images



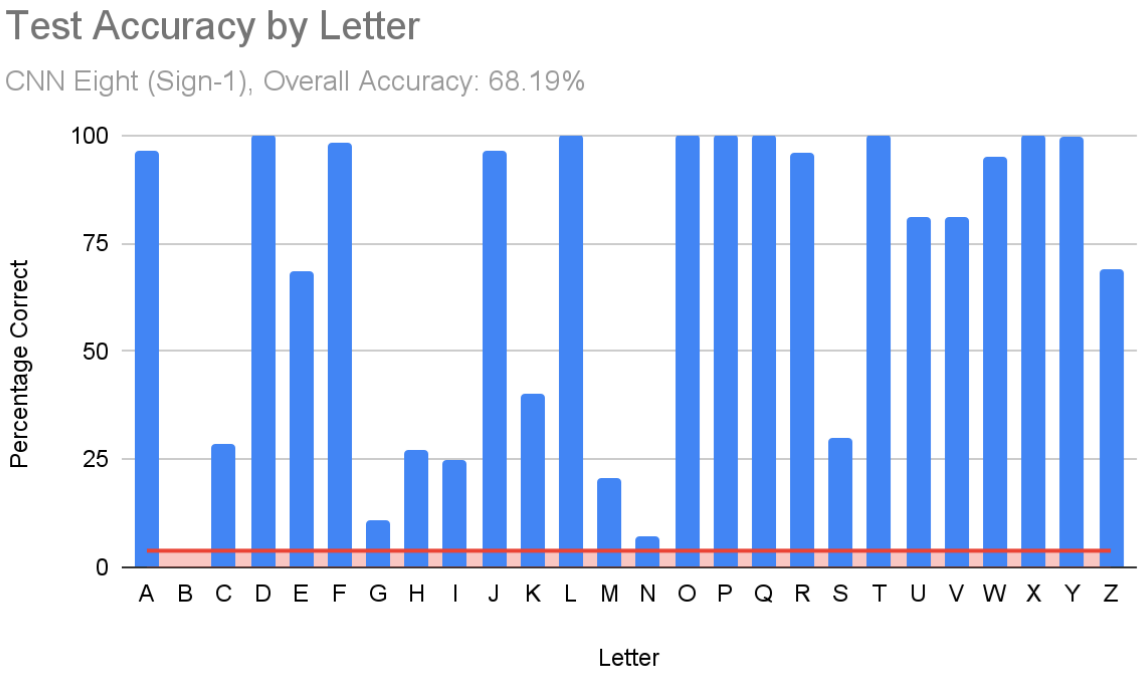


```python
for epoch in range(EPOCHS):
    running_loss = 0.0
    for i, data in enumerate(train_dataloader, 0):
        inputs, labels = data
        optimizer.zero_grad()
        inputs = inputs.float()
        outputs = model(inputs) #.to("cuda")
        outputs #.to("cuda")
        loss = criterion(outputs, labels) #.to("cuda")
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
        if i % 2000 == 1999:
            print(f'[{epoch + 1}, {i + 1:5d}] loss: {running_loss / 2000:.3f}')
            running_loss = 0.0
```

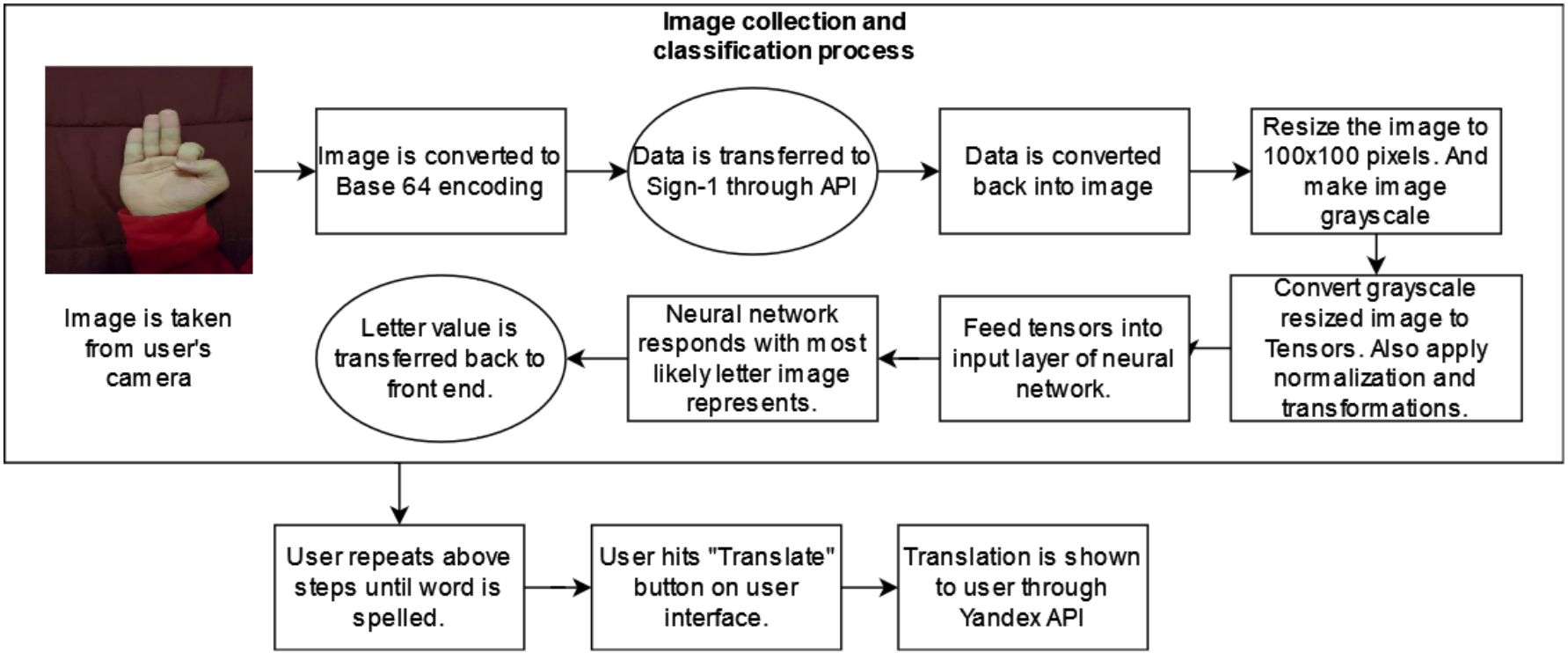
3. Accuracy Graphs



4. Accuracy Graphs

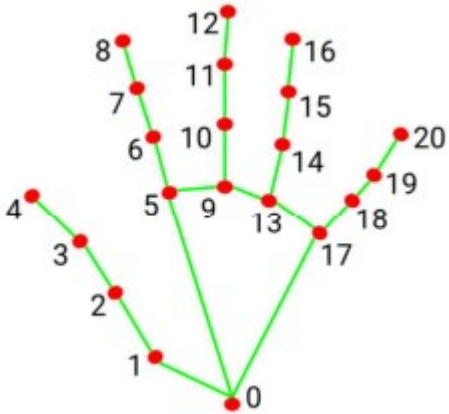


5. Flowchart of Знак-1 (Sign - 1)



VI. Comparison with Similar Research Project

1. Similar problem was researched by two researchers in Novosibirsk, Russia.
2. “Development of a Software Module for Recognizing the Fingerspelling of the Russian Sign Language based on LSTM”
3. Used MediaPipe Hands to create their input data from short videos.
4. Their model was determined to have a 91% accuracy on test data.



VII. Language Data Selection

1. Original models (Sign-0) were trained on data taken from Kaggle.
2. New convolutional models (Sign-1) were trained on self-made dataset, with help of three friends.
3. Self made dataset comprised of 189,632 images. Over 30 gigabytes in size!
4. Dataset created by taking video of American Sign Language letters over a dark background. Then feeding the videos through my data processing scripts.
5. Russian was selected as target language due to familiarity with the language and friends who are native Russian speakers.

VIII. Language Skills

1. Our product tries to translate one area of the four main language skills: Speaking.
2. We expect both of our users to be proficient in Reading and Writing.
3. In later revisions of Znak, the language both users use will not matter.

Language Skills

	Written	**Oral**
Passive	Reading	Listening
Active	Writing	Speaking

IX. Challenges

1. Pytorch documentation was hard to read and made it easy to make mistakes.
2. Some American Sign Language letters are very similar to each other, and some people sign these letters very differently.
3. Debugging neural networks was very difficult and time consuming.
4. Website front end introduced unexpected bugs and complications to the project deployment and demonstration.
5. Image collection and preparation took much more time than anticipated.
6. The usage of image classification and recognition might not be appropriate for real world applications.

X. Future Directions

Знак-2 (Sign-2)

- Use Mediapipe Hands tool to reduce feature set, greatly improving potential accuracy, reducing training time, and decreasing the computational cost of the model immensely.

Знак-3 (Sign-3)

- Add support for video based input using LSTM, allowing word level American Sign Language translation.

Знак-4 (Sign-4)

- Create a pipeline for easily adding additional sign languages, such as Russian Sign Language, International Sign, and more.

Selected Bibliography

American Council of Teachers of Foreign Languages **(ACTFL). Guidelines:** https://www.actfl.org/

Kinsley, Harrison and Daniel Kukiela. *Neural Networks from Scratch*. Self Published, 2020.

M. G. Grif and Y. K. Kondratenko. *Development of a software module for recognizing the fingerspelling of the Russian Sign Language based on LSTM. Journal of Physics: Conference Series*, vol. 2032, no. 012024, May 2021

Oxford, R. *Second language learning strategies: Current research and implications for practice*.

Center for Language Education and Research, 11 June 1987.

Acknowledgements

Dr. Yu-Hsien Chiu - Department of Business and Information Technology, Missouri S&T for Help with project goal refinement and providing early feedback.

Dr. Irina Ivliyeva - Department of Arts, Language, and Philosophy, Missouri S&T for Help with developing research methodology, project organization and planning, evaluation/gathering of language resources, presentation design and delivery tips.

Dr. Audra Merfeld-Langston - Department of Arts, Language, and Philosophy, Missouri S&T - Help with advising during project, keeping me on schedule, and project refinement.

Dr. Langtao Chen - Department of Business and Information Technology, Missouri S&T - Help with understanding classification algorithms and resampling methods for model development.

Mr. Grey Whittenberger - Website front end help and assistance with Heroku deployment for demonstration.

Questions and Discussion
Вопросы и обсуждение

Thank you for your attention and help!
Спасибо за внимание и за помощь!