Instituto Tecnológico de Estudios Superiores de Monterrey

ACTIVIDAD:

“05 - Proyecto Final (CNN)”

Integrantes:

- Bernardo García Zermeño | A00570682
- Omar David Hernández Aguirre | A01383543

Instructor: Luis Ricardo Peña Llamas

14 de junio de 2023

Monterrey, Nuevo León

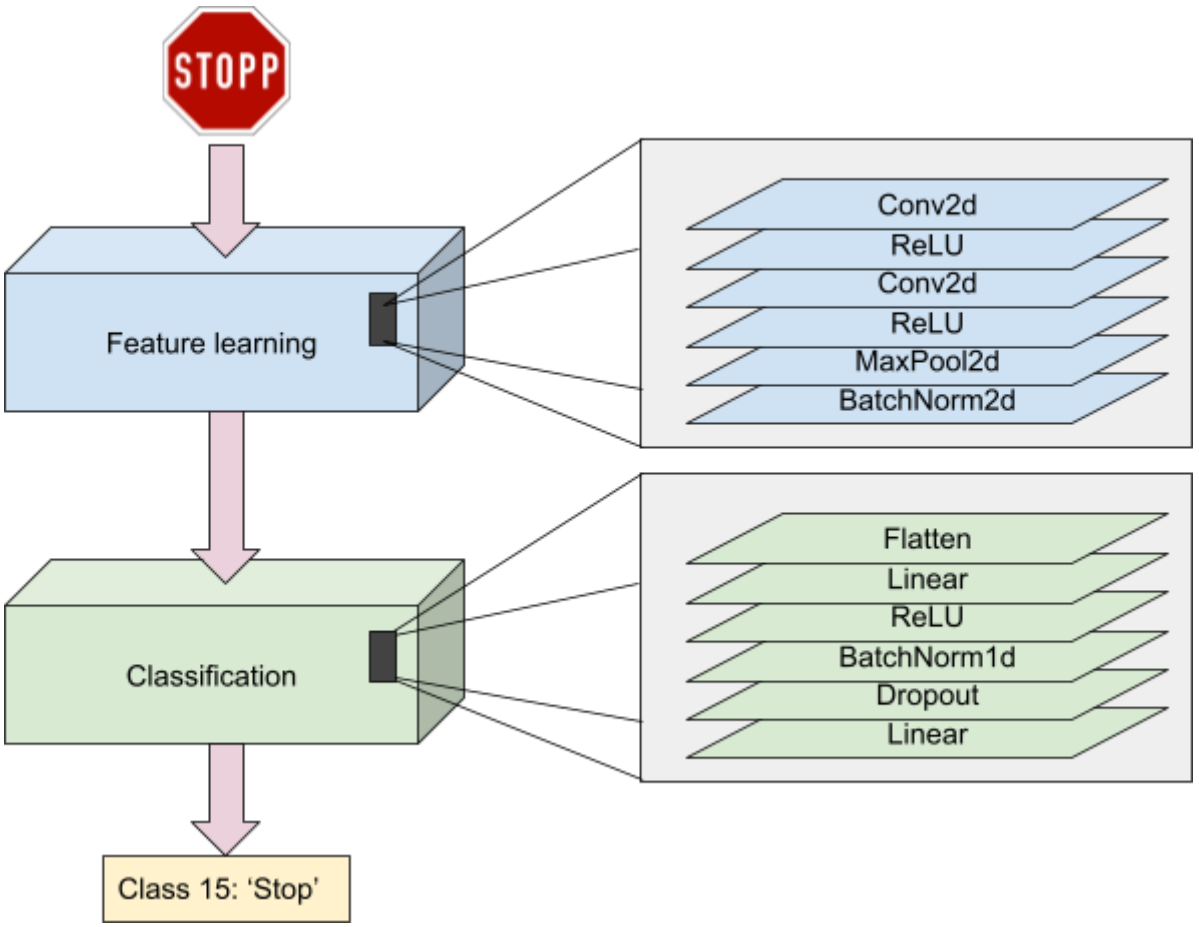
Introducción

El siguiente trabajo describe la red neuronal convolucional que se construyó para clasificar más de 40 clases de señales de tráfico alemanas. Para esto, el modelo se entrenó con un *dataset* de más de 50,000 imágenes llegando a predecir éstas señales de tráfico con un *accuracy* de aproximadamente 65%.

Las siguientes son las clases en las que se clasifican las imágenes:

```
classes = { 1:'Speed limit (20km/h)',
            2:'Speed limit (30km/h)',
            3:'Speed limit (50km/h)',
            4:'Speed limit (60km/h)',
            5:'Speed limit (70km/h)',
            6:'Speed limit (80km/h)',
            7:'End of speed limit (80km/h)',
            8:'Speed limit (100km/h)',
            9:'Speed limit (120km/h)',
            10:'No passing',
            11:'No passing veh over 3.5 tons',
            12:'Right-of-way at intersection',
            13:'Priority road',
            14:'Yield',
            15:'Stop',
            16:'No vehicles',
            17:'Veh > 3.5 tons prohibited',
            18:'No entry',
            19:'General caution',
            20:'Dangerous curve left',
            21:'Dangerous curve right',
            22:'Double curve',
            23:'Bumpy road',
            24:'Slippery road',
            25:'Road narrows on the right',
            26:'Road work',
            27:'Traffic signals',
            28:'Pedestrians',
            29:'Children crossing',
            30:'Bicycles crossing',
            31:'Beware of ice/snow',
            32:'Wild animals crossing',
            33:'End speed + passing limits',
            34:'Turn right ahead',
            35:'Turn left ahead',
            36:'Ahead only',
            37:'Go straight or right',
            38:'Go straight or left',
            39:'Keep right',
            40:'Keep left',
            41:'Roundabout mandatory',
            42:'End of no passing',
            43:'End no passing veh > 3.5 tons' }
```

Arquitectura del modelo



Entrenamiento del modelo

El *dataset* se dividió en lotes 64 y se realizaron 15 *epochs* para alcanzar un ***accuracy* de 63%** y un ***average loss* de 0.08028**.

```
Training - Epoch: 0  Accuracy: 27.808553971486763 Avg Loss: 2.306763921395824
Training - Epoch: 1  Accuracy: 47.78004073319756 Avg Loss: 1.118791427000477
Training - Epoch: 2  Accuracy: 54.936863543788185 Avg Loss: 0.6958316609475861
Training - Epoch: 3  Accuracy: 58.4928716904277 Avg Loss: 0.48317858000151254
Training - Epoch: 4  Accuracy: 60.02851323828921 Avg Loss: 0.36410335619799233
Training - Epoch: 5  Accuracy: 61.14256619144603 Avg Loss: 0.2867897265342248
Training - Epoch: 6  Accuracy: 61.84114052953157 Avg Loss: 0.23512004649687687
Training - Epoch: 7  Accuracy: 62.25661914460285 Avg Loss: 0.1975489134336926
Training - Epoch: 8  Accuracy: 62.55804480651731 Avg Loss: 0.17044316916565302
Training - Epoch: 9  Accuracy: 62.84521384928717 Avg Loss: 0.14763786452547353
Training - Epoch: 10  Accuracy: 62.97759674134419 Avg Loss: 0.130346898930318
Training - Epoch: 11  Accuracy: 63.11812627291243 Avg Loss: 0.11683125449477291
Training - Epoch: 12  Accuracy: 63.193482688391036 Avg Loss: 0.10473448978907708
Training - Epoch: 13  Accuracy: 63.29327902240326 Avg Loss: 0.09511382746569008
Training - Epoch: 14  Accuracy: 63.40936863543788 Avg Loss: 0.08666701412279834
Training - Epoch: 15  Accuracy: 63.39511201629328 Avg Loss: 0.08028819144546015
```

Figura 1. Entrenamiento del modelo. En el *epoch* 0 se alcanzó un *accuracy* de 27% con un *average loss* de 2.23067, y para el *epoch* 15 se alcanzó un *accuracy* de 63.39% con un *average loss* de 0.08028.

Predicciones del modelo

Por último, para hacer el *testing* lo que se hace es tomar el tensor (matriz) de los datos, donde cada espacio en la matriz incluye el número de clase de cada imagen en el batch (ronda de prueba), y lo compara con el tensor que predice nuestra red.

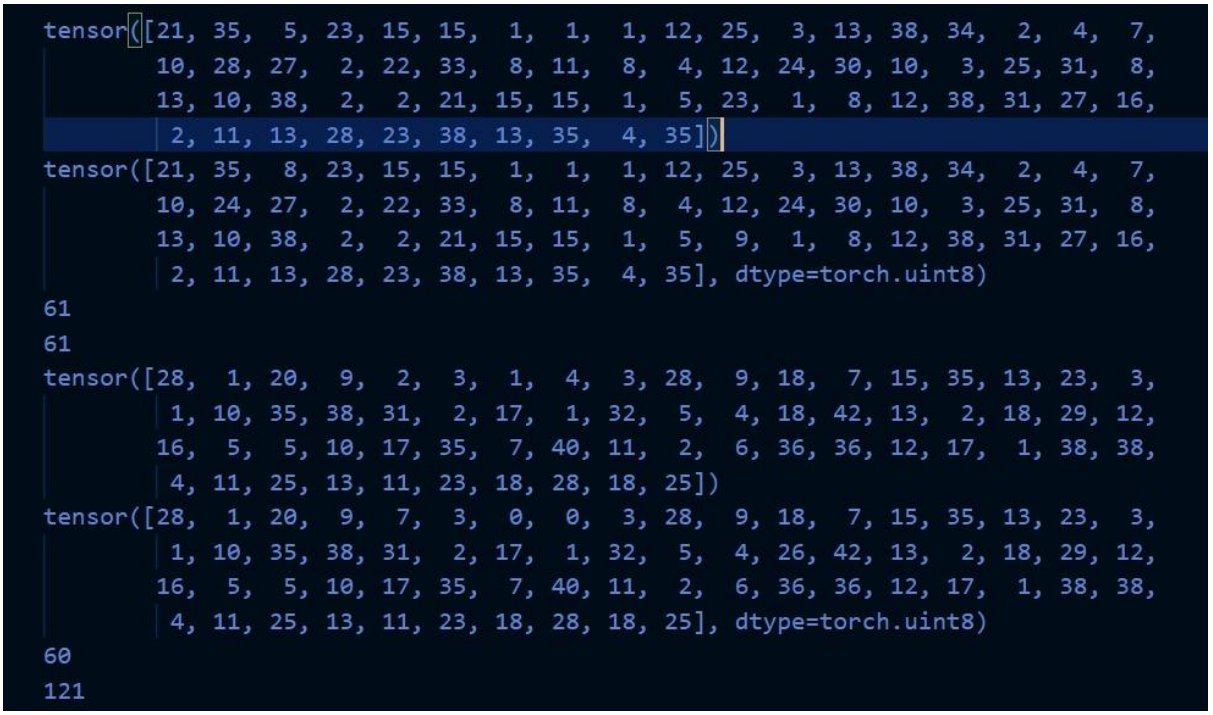


Figura 2. Probando el modelo.

Video

[Link de video]