In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import tensorflow as tf
import matplotlib.pyplot as plt
import sklearn
import os
import plotly
import plotly.graph_objs as go
import time
import itertools
import cv2
import seaborn as sns
import warnings
import tqdm
import math
```

In [2]:

```
warnings.simplefilter(action='ignore', category=FutureWarning)
%matplotlib inline
#get_ipython().run_line_magic('matplotlib', 'inline')
plotly.offline.init_notebook_mode(True)
dataset_dir = './'
meta_info = os.path.join(dataset_dir, 'Meta.csv')
train_csv_path = os.path.join(dataset_dir, 'Train.csv')

test_csv_path = os.path.join(dataset_dir, 'Test.csv')
```

In [3]:

```
labels = ['20 km/h', '30 km/h', '50 km/h', '60 km/h', '70 km/h', '80 km/h', '80 km/h end',
'100 km/h', '120 km/h', 'No overtaking',
'No overtaking for tracks', 'Crossroad with secondary way', 'Main road', 'Give way', 'Stop',
'Road up', 'Road up for track', 'Brock',
'Other dangerous', 'Turn left', 'Turn right', 'Winding road', 'Hollow road', 'Slippery road',
'Narrowing road', 'Roadwork', 'Traffic light',
'Pedestrian', 'Children', 'Bike', 'Snow', 'Deer', 'End of the limits', 'Only right', 'Only left',
'Only straight', 'Only straight and right',
'Only straight and left', 'Take right', 'Take left', 'Circle crossroad', 'End of overtaking limit', 'End of overtaking limit for track']
```

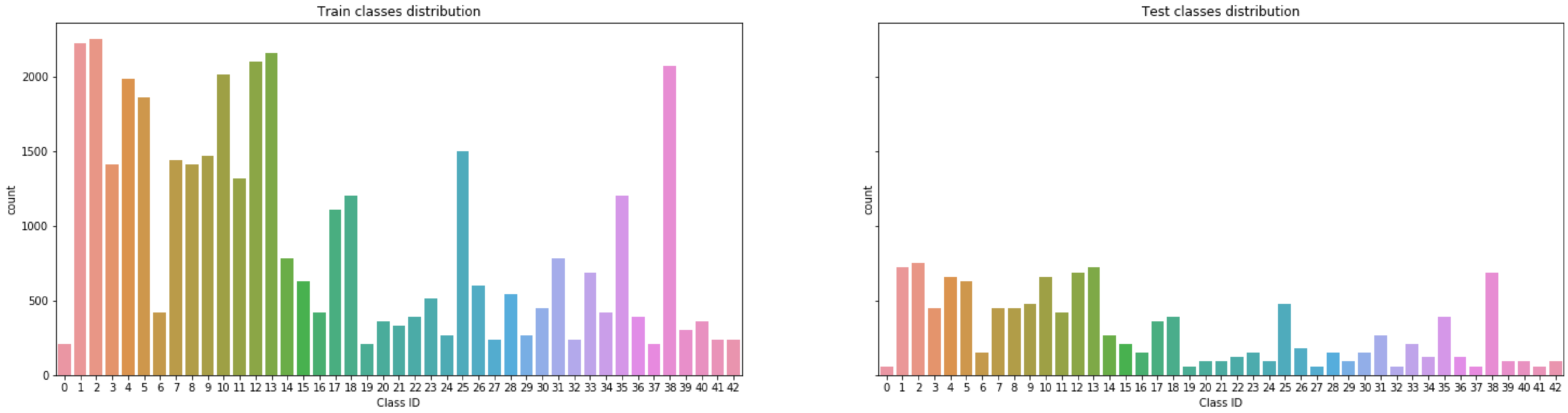
In [4]:

```
train_data_color = '#0f7b8e'
test_data_color = '#630f8e'
trainDf = pd.read_csv(train_csv_path)
testDf = pd.read_csv(test_csv_path)
metaDf = pd.read_csv(meta_info)
trainDf['Path'] = list(map(lambda x: os.path.join(dataset_dir,x.lower()), trainDf['Path']))
testDf['Path'] = list(map(lambda x: os.path.join(dataset_dir,x.lower()), testDf['Path']))
metaDf['Path'] = list(map(lambda x: os.path.join(dataset_dir,x.lower()), metaDf['Path']))
trainDf.sample(3)
```

Out[4]:

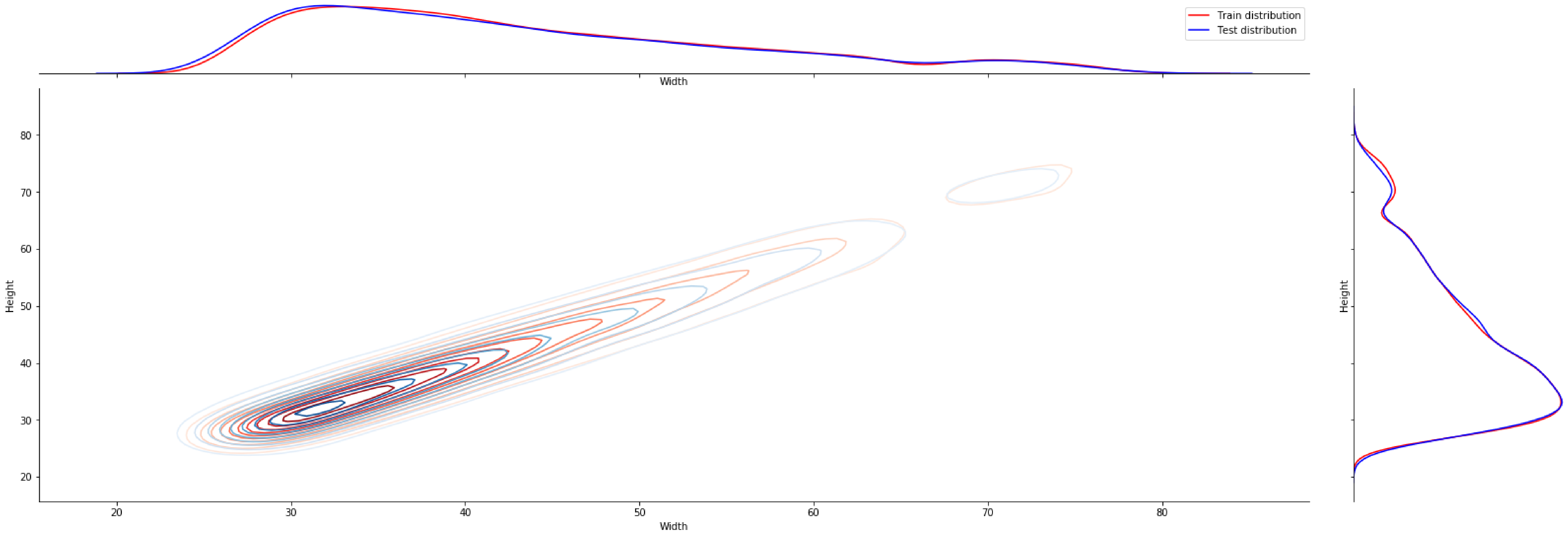
	Width	Height	Roi.X1	Roi.Y1	Roi.X2	Roi.Y2	ClassId	Path
23687	51	52	5	6	46	47	15	./train/15/00015_00009_00017.png
38982	38	39	5	5	32	33	42	./train/42/00042_00000_00013.png
4655	36	36	5	5	31	31	2	./train/2/00002_00062_00005.png

In [5]:
```
fig, axs = plt.subplots(1, 2, sharex=True, sharey=True, figsize=(25, 6))
axs[0].set_title('Train classes distribution')
axs[0].set_xlabel('Class')
axs[0].set_ylabel('Count')
axs[1].set_title('Test classes distribution')
axs[1].set_xlabel('Class')
axs[1].set_ylabel('Count')
sns.countplot(trainDf.ClassId, ax=axs[0])
sns.countplot(testDf.ClassId, ax=axs[1])
axs[0].set_xlabel('Class ID');
axs[1].set_xlabel('Class ID');
```



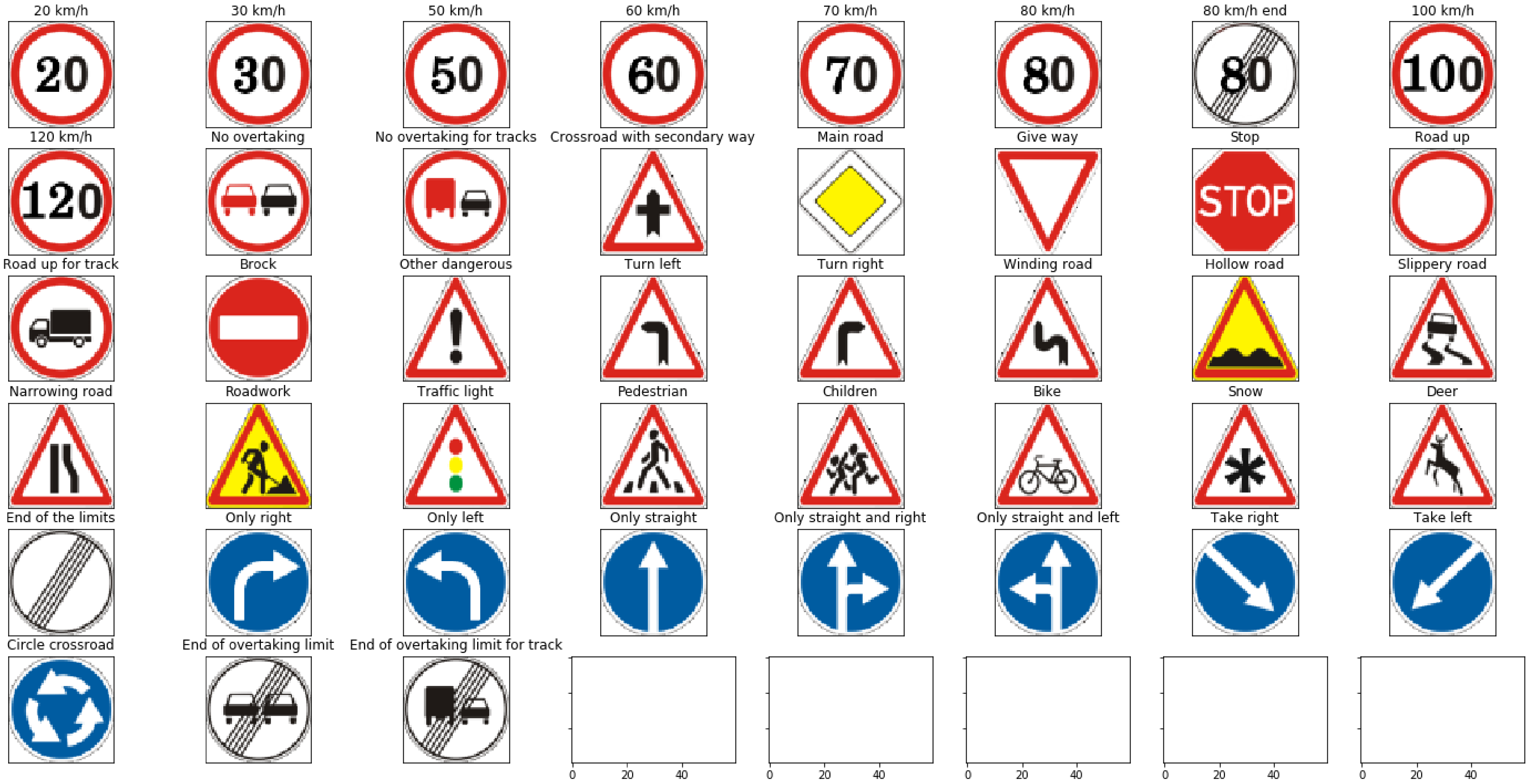
In [6]:

```
trainDfDpiSubset = trainDf[(trainDf.Width < 80) & (trainDf.Height < 80)];
testDfDpiSubset = testDf[(testDf.Width < 80) & (testDf.Height < 80)];
g = sns.JointGrid(x="Width", y="Height", data=trainDfDpiSubset)
sns.kdeplot(trainDfDpiSubset.Width, trainDfDpiSubset.Height, cmap="Reds",
shade=False, shade_lowest=False, ax=g.ax_joint)
sns.kdeplot(testDfDpiSubset.Width, testDfDpiSubset.Height, cmap="Blues",
shade=False, shade_lowest=False, ax=g.ax_joint)
sns.distplot(trainDfDpiSubset.Width, kde=True, hist=False, color="r", ax=g.ax_marg_x,
label='Train distribution')
sns.distplot(testDfDpiSubset.Width, kde=True, hist=False, color="b", ax=g.ax_marg_x,
label='Test distribution')
sns.distplot(trainDfDpiSubset.Width, kde=True, hist=False, color="r", ax=g.ax_marg_y,
vertical=True)
sns.distplot(testDfDpiSubset.Height, kde=True, hist=False, color="b", ax=g.ax_marg_y,
vertical=True)
g.fig.set_figwidth(25)
g.fig.set_figheight(8)
plt.show();
```



In [7]:

```
sns.set_style()
rows = 6
cols = 8
fig, axs = plt.subplots(rows, cols, sharex=True, sharey=True, figsize=(25, 12))
plt.subplots_adjust(left=None, bottom=None, right=None, top=0.9, wspace=None,
hspace=None)
metaDf = metaDf.sort_values(by=['ClassId'])
idx = 0
for i in range(rows):
    for j in range(cols):
        if idx > 42:
            break
        img = cv2.imread(metaDf["Path"].tolist()[idx], cv2.IMREAD_UNCHANGED)
        img[np.where(img[:,:,3]==0)] = [255,255,255,255]
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (60,60))
        axs[i,j].imshow(img)
        axs[i,j].set_facecolor('xkcd:salmon')
        axs[i,j].set_facecolor((1.0, 0.47, 0.42))
        axs[i,j].set_title(labels[int(metaDf["ClassId"].tolist()[idx])])
        axs[i,j].get_xaxis().set_visible(False)
        axs[i,j].get_yaxis().set_visible(False)
        idx += 1
```


20 km/h
30 km/h
50 km/h
60 km/h
70 km/h
80 km/h
80 km/h end
100 km/h
120 km/h
No overtaking
No overtaking for tracks
Crossroad with secondary way
Main road
Give way
Stop
Road up
Road up for track
Brock
Other dangerous
Turn left
Turn right
Winding road
Hollow road
Slippery road
Narrowing road
Roadwork
Traffic light
Pedestrian
Children
Bike
Snow
Deer
End of the limits
Only right
Only left
Only straight
Only straight and right
Only straight and left
Take right
Take left
Circle crossroad
End of overtaking limit
End of overtaking limit for track
0
20
40
0
20
40
0
20
40
0
20
40
0
20
40

In [8]:
```
rows = 6
cols = 8+4
fig, axs = plt.subplots(rows, cols, sharex=True, sharey=True, figsize=(25, 12))
plt.subplots_adjust(left=None, bottom=None, right=None, top=0.9, wspace=None,
hspace=None)
visualize = trainDf.sample(rows*cols)
idx = 0
for i in range(rows):
    for j in range(cols):
        img = cv2.imread(visualize["Path"].tolist()[idx])
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (60,60))
        axs[i,j].imshow(img)
        axs[i,j].set_title(labels[int(visualize["ClassId"].tolist()[idx])])
        axs[i,j].get_xaxis().set_visible(False)
        axs[i,j].get_yaxis().set_visible(False)
        idx += 1
```

In [9]:

```
img_load_size = (60,60)
zero_img = np.zeros([12,img_load_size[0], img_load_size[1], 3])
zero_label = np.zeros([12,1])
def parse_function(filename, label):
    image_string = tf.read_file(filename)
    image = tf.image.decode_jpeg(image_string, channels=3)
    # image = tf.py_func(eq, [image], image.dtype)52
    image.set_shape([None, None, 3])
    return filename, image, label
def train_preprocess(filename, image, label):
    image = tf.image.convert_image_dtype(image, tf.float32)
    image = tf.image.resize_images(image, img_load_size)
    return filename, image, label
def augmentate(filename, image, label):
    grad = tf.random.uniform(shape=[], minval=-0.3, maxval=0.3)
    dx = tf.random.uniform(shape=[], minval=-15, maxval=15, dtype=tf.int32)
    dy = tf.random.uniform(shape=[], minval=-15, maxval=15, dtype=tf.int32)
    image = tf.contrib.image.rotate(image, grad)
    image = tf.contrib.image.translate(image, translations=[dx, dy])
    return filename, image, label
def eq(img: np.ndarray):
    res = img.copy()
    res[:, :, 0] = cv2.equalizeHist(img[:, :, 0])
    res[:, :, 1] = cv2.equalizeHist(img[:, :, 1])
    res[:, :, 2] = cv2.equalizeHist(img[:, :, 2])
    return res53
def tf_equalize_histogram(image):
    values_range = tf.constant([0., 255.], dtype = tf.float32)
    histogram = tf.histogram_fixed_width(tf.to_float(image), values_range, 256)
    cdf = tf.cumsum(histogram)
    cdf_min = cdf[tf.reduce_min(tf.where(tf.greater(cdf, 0)))]
    img_shape = tf.shape(image)
    pix_cnt = img_shape[-3] * img_shape[-2]
    px_map = tf.round(tf.to_float(cdf - cdf_min) * 255. / tf.to_float(pix_cnt - 1))
    px_map = tf.cast(px_map, tf.uint8)
    gth = tf.gather_nd(px_map, tf.cast(image, tf.int32))
    eq_hist = tf.expand_dims(gth, 2)
    return image
```

In [10]:

```
print(tf.__version__)
```

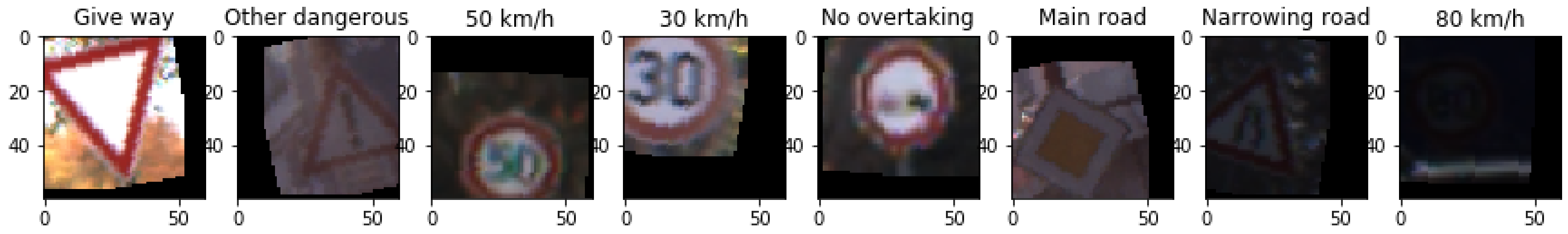
```
1.14.0
```

In [11]:

```
tf.reset_default_graph()
epochs = 100
batch_size = 12
prefetch_count = 1
samples_train = len(trainDf)
samples_test = len(testDf)
dataset_train = tf.data.Dataset.from_tensor_slices((trainDf['Path'], trainDf['ClassId']))
dataset_train = dataset_train.shuffle(len(trainDf['Path']))
dataset_train = dataset_train.repeat(epochs)
dataset_train = dataset_train.map(parse_function, num_parallel_calls=4)
dataset_train = dataset_train.map(train_preprocess, num_parallel_calls=4)
dataset_train = dataset_train.map(augmentate, num_parallel_calls=4)
dataset_train = dataset_train.batch(batch_size)
dataset_train = dataset_train.prefetch(prefetch_count)
dataset_iterator = tf.compat.v1.data.Iterator.from_structure(dataset_train.output_types,
dataset_train.output_shapes)
dataset_test = tf.data.Dataset.from_tensor_slices((testDf['Path'], testDf['ClassId']))
dataset_test = dataset_test.shuffle(len(testDf['Path']))
dataset_test = dataset_test.repeat(epochs+1)
dataset_test = dataset_test.map(parse_function, num_parallel_calls=4)
dataset_test = dataset_test.map(train_preprocess, num_parallel_calls=4)
dataset_test = dataset_test.batch(batch_size)
dataset_test = dataset_test.prefetch(prefetch_count)
train_init_op = dataset_iterator.make_initializer(dataset_train)
test_init_op = dataset_iterator.make_initializer(dataset_test)
load_filename, load_img, load_label = dataset_iterator.get_next()
```

```
WARNING:tensorflow:
The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md
  * https://github.com/tensorflow/addons
  * https://github.com/tensorflow/io (for I/O related ops)
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From <ipython-input-11-3da82a42fdda>:15: DatasetV1.output_types (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be removed in a future ver
sion.
Instructions for updating:
Use `tf.compat.v1.data.get_output_types(dataset)`.
WARNING:tensorflow:From <ipython-input-11-3da82a42fdda>:16: DatasetV1.output_shapes (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be removed in a future ve
rsion.
Instructions for updating:
Use `tf.compat.v1.data.get_output_shapes(dataset)`.
WARNING:tensorflow:From C:\Users\vkkre\anaconda3\lib\site-packages\tensorflow\python\data\ops\iterator_ops.py:348: Iterator.output_types (from tensorflow.python.data.ops.iterator
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_types(iterator)`.
WARNING:tensorflow:From C:\Users\vkkre\anaconda3\lib\site-packages\tensorflow\python\data\ops\iterator_ops.py:349: Iterator.output_shapes (from tensorflow.python.data.ops.iterato
r_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_shapes(iterator)`.
WARNING:tensorflow:From C:\Users\vkkre\anaconda3\lib\site-packages\tensorflow\python\data\ops\iterator_ops.py:351: Iterator.output_classes (from tensorflow.python.data.ops.iterat
or_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_classes(iterator)`.
```

In [12]:
```
fig, ax = plt.subplots(ncols=8, nrows=1, figsize=(15, 6))
with tf.Session() as sess:
    sess.run(train_init_op)
    for j in range(8):
        i, l = sess.run([load_img, load_label])
        i = (i[0]*255).astype(np.uint8)
        ax[j].imshow(i)
        ax[j].set_title(labels[l[0]])
```



In [13]:
```
dp_rate = tf.placeholder(dtype=tf.float32, shape=[], name='dp_rate')
img_placeholder = tf.placeholder(shape=[None, 60,60,3], dtype=tf.float32,
name='img_placeholder')
label_placeholder = tf.placeholder(shape=[None, 1], dtype=tf.int64,
name='label_placeholder')
manual_load = tf.placeholder(dtype=tf.bool, shape=[], name='manual_load_placeholder')
# inp = net = tf.cond(pred=manual_load, true_fn=lambda : img_placeholder, false_fn=lambda : load_img, name='network_start'
# label = tf.cond(pred=manual_load, true_fn=lambda : label_placeholder, false_fn=lambda : load_label, name='label'
inp = net = tf.cond(manual_load, lambda: img_placeholder, lambda: load_img)
label = load_label
conv1 = net = tf.layers.conv2d(inputs=net, filters=16, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
conv2 = net = tf.layers.conv2d(inputs=net, filters=32, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
conv3 = net = tf.layers.conv2d(inputs=net, filters=32, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
conv4 = net = tf.layers.conv2d(inputs=net, filters=64, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
net = tf.layers.max_pooling2d(inputs=net, pool_size=(2,2), strides=(2,2))
conv5 = net = tf.layers.conv2d(inputs=net, filters=64, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
conv6 = net = tf.layers.conv2d(inputs=net, filters=128, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
conv5 = net = tf.layers.conv2d(inputs=net, filters=256, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
conv6 = net = tf.layers.conv2d(inputs=net, filters=400, kernel_size=(3,3), strides=(1,1), activation=tf.nn.leaky_relu)
net = tf.layers.batch_normalization(inputs=net)
flatten1 = net = tf.layers.flatten(inputs=net)
dp1 = net = tf.layers.dropout(inputs=net, rate=dp_rate)
dense1 = net = tf.layers.dense(inputs=net, units=256)
logits = tf.layers.dense(inputs=net, units=43)
pred_classes = tf.argmax(logits, axis=1)
pred_probas = tf.nn.softmax(logits)
acc, acc_op = tf.metrics.accuracy(labels=label, predictions=pred_classes)
end_loss = tf.losses.sparse_softmax_cross_entropy(logits=logits, labels=label)
loss = end_loss
label_transpose = tf.transpose(label)
correct_prediction = tf.equal(pred_classes, label_transpose)
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
confusion_matrix_op = tf.confusion_matrix(labels=label, predictions=pred_classes, num_classes=43)
opt = tf.train.AdamOptimizer(learning_rate=0.0001).minimize(loss)
```

```
WARNING:tensorflow:From <ipython-input-13-c2bd81117dd1>:11: conv2d (from tensorflow.python.layers.convolutional) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.keras.layers.Conv2D` instead.
WARNING:tensorflow:From C:\Users\vkkre\anaconda3\lib\site-packages\tensorflow\python\ops\init_ops.py:1251: calling VarianceScaling.__init__ (from tensorflow.python.ops.init_ops)
with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD4181C8>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD4181C8>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD4181C8>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD4181C8>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:From <ipython-input-13-c2bd81117dd1>:12: batch_normalization (from tensorflow.python.layers.normalization) is deprecated and will be removed in a future versio
n.
Instructions for updating:
Use keras.layers.BatchNormalization instead.  In particular, `tf.control_dependencies(tf.GraphKeys.UPDATE_OPS)` should not be used (consult the `tf.keras.layers.batch_normalizati
on` documentation).
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D9B48>> could not be transfor
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D9B48>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D9B48>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D9B48>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3748>> could not be transfor
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3748>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3748>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3748>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD198F88>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD198F88>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD198F88>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD198F88>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>> could not be transfor
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3748>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3748>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3748>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3748>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>> could not be transfor
```

```
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD3D3948>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:From <ipython-input-13-c2bd81117dd1>:19: max_pooling2d (from tensorflow.python.layers.pooling) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.MaxPooling2D instead.
WARNING:tensorflow:Entity <bound method Pooling2D.call of <tensorflow.python.layers.pooling.MaxPooling2D object at 0x00000239AD2A5A48>> could not be transformed and will be execu
ted as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause:
converting <bound method Pooling2D.call of <tensorflow.python.layers.pooling.MaxPooling2D object at 0x00000239AD2A5A48>>: AssertionError: Bad argument number for Name: 3, expecti
ng 4
WARNING: Entity <bound method Pooling2D.call of <tensorflow.python.layers.pooling.MaxPooling2D object at 0x00000239AD2A5A48>> could not be transformed and will be executed as-is.
Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting
<bound method Pooling2D.call of <tensorflow.python.layers.pooling.MaxPooling2D object at 0x00000239AD2A5A48>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2A5C08>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2A5C08>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2A5C08>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2A5C08>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD2A5A48>> could not be transfor
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD2A5A48>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD2A5A48>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD2A5A48>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2CF708>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2CF708>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2CF708>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD2CF708>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>> could not be transfor
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD268B88>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD268B88>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD268B88>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD268B88>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>> could not be transfor
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>> could not be transformed and will be executed a
s-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: conve
```

```
rting <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>> could not be transformed and will be executed as-is. Plea
se report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bou
nd method Conv.call of <tensorflow.python.layers.convolutional.Conv2D object at 0x00000239AD3D3948>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>> could not be transfor
med and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the
full output. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>>: AssertionError:
Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>> could not be transformed and wi
ll be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full outp
ut. Cause: converting <bound method BatchNormalization.call of <tensorflow.python.layers.normalization.BatchNormalization object at 0x00000239AD1F9CC8>>: AssertionError: Bad argu
ment number for Name: 3, expecting 4
WARNING:tensorflow:From <ipython-input-13-c2bd81117dd1>:28: flatten (from tensorflow.python.layers.core) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.flatten instead.
WARNING:tensorflow:Entity <bound method Flatten.call of <tensorflow.python.layers.core.Flatten object at 0x00000239AD2A5A48>> could not be transformed and will be executed as-is.
Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting
<bound method Flatten.call of <tensorflow.python.layers.core.Flatten object at 0x00000239AD2A5A48>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING: Entity <bound method Flatten.call of <tensorflow.python.layers.core.Flatten object at 0x00000239AD2A5A48>> could not be transformed and will be executed as-is. Please re
port this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound me
thod Flatten.call of <tensorflow.python.layers.core.Flatten object at 0x00000239AD2A5A48>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING:tensorflow:From <ipython-input-13-c2bd81117dd1>:29: dropout (from tensorflow.python.layers.core) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.dropout instead.
WARNING:tensorflow:Entity <bound method Dropout.call of <tensorflow.python.layers.core.Dropout object at 0x00000239AD2CF708>> could not be transformed and will be executed as-is.
Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting
<bound method Dropout.call of <tensorflow.python.layers.core.Dropout object at 0x00000239AD2CF708>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING: Entity <bound method Dropout.call of <tensorflow.python.layers.core.Dropout object at 0x00000239AD2CF708>> could not be transformed and will be executed as-is. Please re
port this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound me
thod Dropout.call of <tensorflow.python.layers.core.Dropout object at 0x00000239AD2CF708>>: AssertionError: Bad argument number for Name: 3, expecting 4
WARNING:tensorflow:From <ipython-input-13-c2bd81117dd1>:30: dense (from tensorflow.python.layers.core) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.dense instead.
WARNING:tensorflow:Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2A5A48>> could not be transformed and will be executed as-is. Ple
ase report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bo
und method Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2A5A48>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING: Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2A5A48>> could not be transformed and will be executed as-is. Please report
this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound method
Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2A5A48>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING:tensorflow:Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2CF708>> could not be transformed and will be executed as-is. Ple
ase report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bo
und method Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2CF708>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING: Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2CF708>> could not be transformed and will be executed as-is. Please report
this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound method
Dense.call of <tensorflow.python.layers.core.Dense object at 0x00000239AD2CF708>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING:tensorflow:From C:\Users\vkkre\anaconda3\lib\site-packages\tensorflow\python\ops\losses\losses_impl.py:121: add_dispatch_support.<locals>.wrapper (from tensorflow.python.
ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

In [14]:

```
config = tf.ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.4
sess = tf.Session(config=config)
saver = tf.train.Saver()
sess.run(tf.global_variables_initializer())
# irn.load_weights('inception_resnet_v2_weights_tf_dim_ordering_tf_kernels_notop.h5')
train_history = {'loss':[], 'acc':[], 'val_loss':[], 'val_acc':[]}
best_acc = 0
for e in range(epochs):
    epoch_history = {'loss':[], 'acc':[], 'val_loss':[], 'val_acc':[]}
    sess.run(train_init_op)
    for i in tqdm.tqdm_notebook(range(samples_train//batch_size), ascii=True, desc='Train epoch {}'.format(e)):
        _, _loss, _acc, mn = sess.run([opt, loss, accuracy, inp], feed_dict={dp_rate: 0.3, manual_load: False, img_placeholder: zero_img, label_placeholder: zero_label})
        # print(np.mean(mn))
        epoch_history['loss'].append(_loss)
        epoch_history['acc'].append(_acc)
    sess.run(test_init_op)
    for i in tqdm.tqdm_notebook(range(samples_test//batch_size), ascii=True, desc='Test epoch {}'.format(e)):
        _loss, _acc = sess.run([loss, accuracy], feed_dict={dp_rate: 0, manual_load: False, img_placeholder: zero_img, label_placeholder: zero_label})
        epoch_history['val_loss'].append(_loss)
        epoch_history['val_acc'].append(_acc)
    train_history['loss'].append(np.mean(epoch_history['loss']))
    train_history['acc'].append(np.mean(epoch_history['acc']))
    train_history['val_loss'].append(np.mean(epoch_history['val_loss']))
    train_history['val_acc'].append(np.mean(epoch_history['val_acc']))
    print("***EPOCH SUMMARY*** Loss: {} Acc: {} | Test Loss: {} Test Acc{}".format(train_history['loss'][-1], train_history['acc'][-1],  train_history['val_loss'][-1], train_hist
ory['val_acc'][-1]))
    if train_history['val_acc'][-1] > best_acc:
        best_acc = train_history['val_acc'][-1]
        save_path = saver.save(sess, "./model.ckpt")
        print("Model saved in path: %s" % save_path)
```

```
C:\Users\vkkre\anaconda3\lib\site-packages\ipykernel_launcher.py:12: TqdmDeprecationWarning:

This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`



C:\Users\vkkre\anaconda3\lib\site-packages\ipykernel_launcher.py:18: TqdmDeprecationWarning:

This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`



***EPOCH SUMMARY*** Loss: 2.034796953201294 Acc: 0.39970409870147705 | Test Loss: 0.9514126181602478 Test Acc0.7100760340690613
Model saved in path: ./model.ckpt



***EPOCH SUMMARY*** Loss: 0.6796357035636902 Acc: 0.7864248752593994 | Test Loss: 0.37691837549209595 Test Acc0.8937737345695496
Model saved in path: ./model.ckpt



***EPOCH SUMMARY*** Loss: 0.3156002163887024 Acc: 0.903708815574646 | Test Loss: 0.31901127099990845 Test Acc0.9204689264297485
Model saved in path: ./model.ckpt



***EPOCH SUMMARY*** Loss: 0.20936134457588196 Acc: 0.9363585710525513 | Test Loss: 0.3752560615539551 Test Acc0.9099335074424744



***EPOCH SUMMARY*** Loss: 0.16352683305740356 Acc: 0.9520202279090881 | Test Loss: 0.3398861289024353 Test Acc0.9345691204071045
Model saved in path: ./model.ckpt



***EPOCH SUMMARY*** Loss: 0.1295175701379776 Acc: 0.9610243439674377 | Test Loss: 0.16924147307872772 Test Acc0.9587294459342957
Model saved in path: ./model.ckpt



***EPOCH SUMMARY*** Loss: 0.1157863438129425 Acc: 0.9668911695480347 | Test Loss: 0.17254915833473206 Test Acc0.9615811109542847
Model saved in path: ./model.ckpt



***EPOCH SUMMARY*** Loss: 0.10227091610431671 Acc: 0.971048891544342 | Test Loss: 0.5121506452560425 Test Acc0.9152407646179199



***EPOCH SUMMARY*** Loss: 0.09174541383981705 Acc: 0.9749515652656555 | Test Loss: 0.1409069299697876 Test Acc0.9688688516616821
Model saved in path: ./model.ckpt



***EPOCH SUMMARY*** Loss: 0.08752381056547165 Acc: 0.9762014150619507 | Test Loss: 0.17828483879566193 Test Acc0.9584125280380249



***EPOCH SUMMARY*** Loss: 0.07834972441196442 Acc: 0.9784461259841919 | Test Loss: 0.16758088767528534 Test Acc0.9671261310577393
```

```
***EPOCH SUMMARY*** Loss: 0.08046724647283554 Acc: 0.9795429110527039 | Test Loss: 0.17762823402881622 Test Acc0.9664924740791321


***EPOCH SUMMARY*** Loss: 0.06682800501585007 Acc: 0.9825528264045715 | Test Loss: 0.15217140316963196 Test Acc0.9652249813079834


***EPOCH SUMMARY*** Loss: 0.06429020315408707 Acc: 0.9834455847740173 | Test Loss: 0.24670523405075073 Test Acc0.9555608630180359


***EPOCH SUMMARY*** Loss: 0.06371679157018661 Acc: 0.9834966063499451 | Test Loss: 0.13926400244235992 Test Acc0.9742554426193237
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.06449111551046371 Acc: 0.9841088056564331 | Test Loss: 0.12065315991640091 Test Acc0.9724335074424744


***EPOCH SUMMARY*** Loss: 0.06422065943479538 Acc: 0.9849505424499512 | Test Loss: 0.2737477719783783 Test Acc0.9631654024124146


***EPOCH SUMMARY*** Loss: 0.06104016304016113 Acc: 0.985460638999939 | Test Loss: 0.17061443626880646 Test Acc0.970453143119812


***EPOCH SUMMARY*** Loss: 0.05971638485789299 Acc: 0.9855881333351135 | Test Loss: 0.08049865067005157 Test Acc0.9809094667434692
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.05363939702510834 Acc: 0.986685037612915 | Test Loss: 0.1675184965133667 Test Acc0.970611572265625


***EPOCH SUMMARY*** Loss: 0.048151422291994095 Acc: 0.9884961247444153 | Test Loss: 0.15188132226467133 Test Acc0.9744930863380432


***EPOCH SUMMARY*** Loss: 0.06063399091362953 Acc: 0.9861749410629272 | Test Loss: 0.14487671852111816 Test Acc0.977820098400116


***EPOCH SUMMARY*** Loss: 0.04881761223077774 Acc: 0.9889807105064392 | Test Loss: 0.1000901609659195 Test Acc0.9792459607124329


***EPOCH SUMMARY*** Loss: 0.05114263296127319 Acc: 0.9886491298675537 | Test Loss: 0.16341634094715118 Test Acc0.9758397340774536


***EPOCH SUMMARY*** Loss: 0.04929023236036301 Acc: 0.9890572428703308 | Test Loss: 0.11899857968091965 Test Acc0.9829689860343933
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.04809577018022537 Acc: 0.9885215759277344 | Test Loss: 0.2816813588142395 Test Acc0.9707699418067932
```

```
***EPOCH SUMMARY*** Loss: 0.04529208317399025 Acc: 0.9896184802055359 | Test Loss: 0.1632981151342392 Test Acc0.9832859039306641
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.04742809012532234 Acc: 0.9893122911453247 | Test Loss: 0.2101684957742691 Test Acc0.9702154397964478


***EPOCH SUMMARY*** Loss: 0.04227793961763382 Acc: 0.9906131625175476 | Test Loss: 0.3290384113788605 Test Acc0.946768045425415


***EPOCH SUMMARY*** Loss: 0.04799986630678177 Acc: 0.9895673990249634 | Test Loss: 0.1407667100429535 Test Acc0.9749682545661926


***EPOCH SUMMARY*** Loss: 0.04104636609554291 Acc: 0.9909957647323608 | Test Loss: 0.11185930669307709 Test Acc0.9786913990974426


***EPOCH SUMMARY*** Loss: 0.04277487099170685 Acc: 0.9908682107925415 | Test Loss: 0.15297269821166992 Test Acc0.9807509779930115


***EPOCH SUMMARY*** Loss: 0.04283709451556206 Acc: 0.9909958839416504 | Test Loss: 0.19431611895561218 Test Acc0.9708492159843445


***EPOCH SUMMARY*** Loss: 0.03892076015472412 Acc: 0.992169201374054 | Test Loss: 0.11717698723077774 Test Acc0.9820184111595154


***EPOCH SUMMARY*** Loss: 0.04265632480382919 Acc: 0.9912763833999634 | Test Loss: 0.1829908788204193 Test Acc0.976314902305603


***EPOCH SUMMARY*** Loss: 0.0413564033806324 Acc: 0.9912763833999634 | Test Loss: 0.12745219469070435 Test Acc0.9809094667434692


***EPOCH SUMMARY*** Loss: 0.041702304035425186 Acc: 0.9918885827064514 | Test Loss: 0.24991418421268463 Test Acc0.9676806330680847


***EPOCH SUMMARY*** Loss: 0.03654766082763672 Acc: 0.9925262331962585 | Test Loss: 0.3272826671600342 Test Acc0.9686312079429626


***EPOCH SUMMARY*** Loss: 0.03824079409241676 Acc: 0.9914039373397827 | Test Loss: 0.17150211334228516 Test Acc0.9751266837120056


***EPOCH SUMMARY*** Loss: 0.03947535902261734 Acc: 0.9920926690101624 | Test Loss: 0.16095201671123505 Test Acc0.9766318798065186


***EPOCH SUMMARY*** Loss: 0.03901690989732742 Acc: 0.9916845560073853 | Test Loss: 0.1780981868505478 Test Acc0.980988621711731
```

```
***EPOCH SUMMARY*** Loss: 0.034910380840301514 Acc: 0.9932405352592468 | Test Loss: 0.1060745045542717 Test Acc0.9851077795028687
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.039111193269491196 Acc: 0.9920416474342346 | Test Loss: 0.38093823194503784 Test Acc0.9573827385902405


***EPOCH SUMMARY*** Loss: 0.036644235253334045 Acc: 0.9931129217147827 | Test Loss: 0.18437336385250092 Test Acc0.980038046836853


***EPOCH SUMMARY*** Loss: 0.038030266761779785 Acc: 0.992781400680542 | Test Loss: 0.16972538828849792 Test Acc0.9791666865348816


***EPOCH SUMMARY*** Loss: 0.03498274087905884 Acc: 0.9928833842277527 | Test Loss: 0.11423196643590927 Test Acc0.9816222786903381


***EPOCH SUMMARY*** Loss: 0.036139506846666336 Acc: 0.9922202229499817 | Test Loss: 0.17947837710380554 Test Acc0.9820182919502258


***EPOCH SUMMARY*** Loss: 0.037579648196697235 Acc: 0.9929089546203613 | Test Loss: 0.18151941895484924 Test Acc0.9762357473373413


***EPOCH SUMMARY*** Loss: 0.03395066410303116 Acc: 0.9930364489555359 | Test Loss: 0.17302532494068146 Test Acc0.9802756905555725


***EPOCH SUMMARY*** Loss: 0.03238876163959503 Acc: 0.9934446215629578 | Test Loss: 0.11335211247205734 Test Acc0.9855829477310181
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.03619879484176636 Acc: 0.9929854869842529 | Test Loss: 0.16343097388744354 Test Acc0.9794836044311523


***EPOCH SUMMARY*** Loss: 0.03131558373570442 Acc: 0.994056761264801 | Test Loss: 0.13735753297805786 Test Acc0.9821767807006836


***EPOCH SUMMARY*** Loss: 0.03365789353847504 Acc: 0.9929089546203613 | Test Loss: 0.21598203480243683 Test Acc0.9746515154838562


***EPOCH SUMMARY*** Loss: 0.03256349265575409 Acc: 0.9938527941703796 | Test Loss: 0.1674850881099701 Test Acc0.9759981036186218


***EPOCH SUMMARY*** Loss: 0.035904984921216965 Acc: 0.9940056800842285 | Test Loss: 0.07365401834249496 Test Acc0.9885931611061096
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.030997877940535545 Acc: 0.9940056800842285 | Test Loss: 0.20506756007671356 Test Acc0.9729087352752686
```

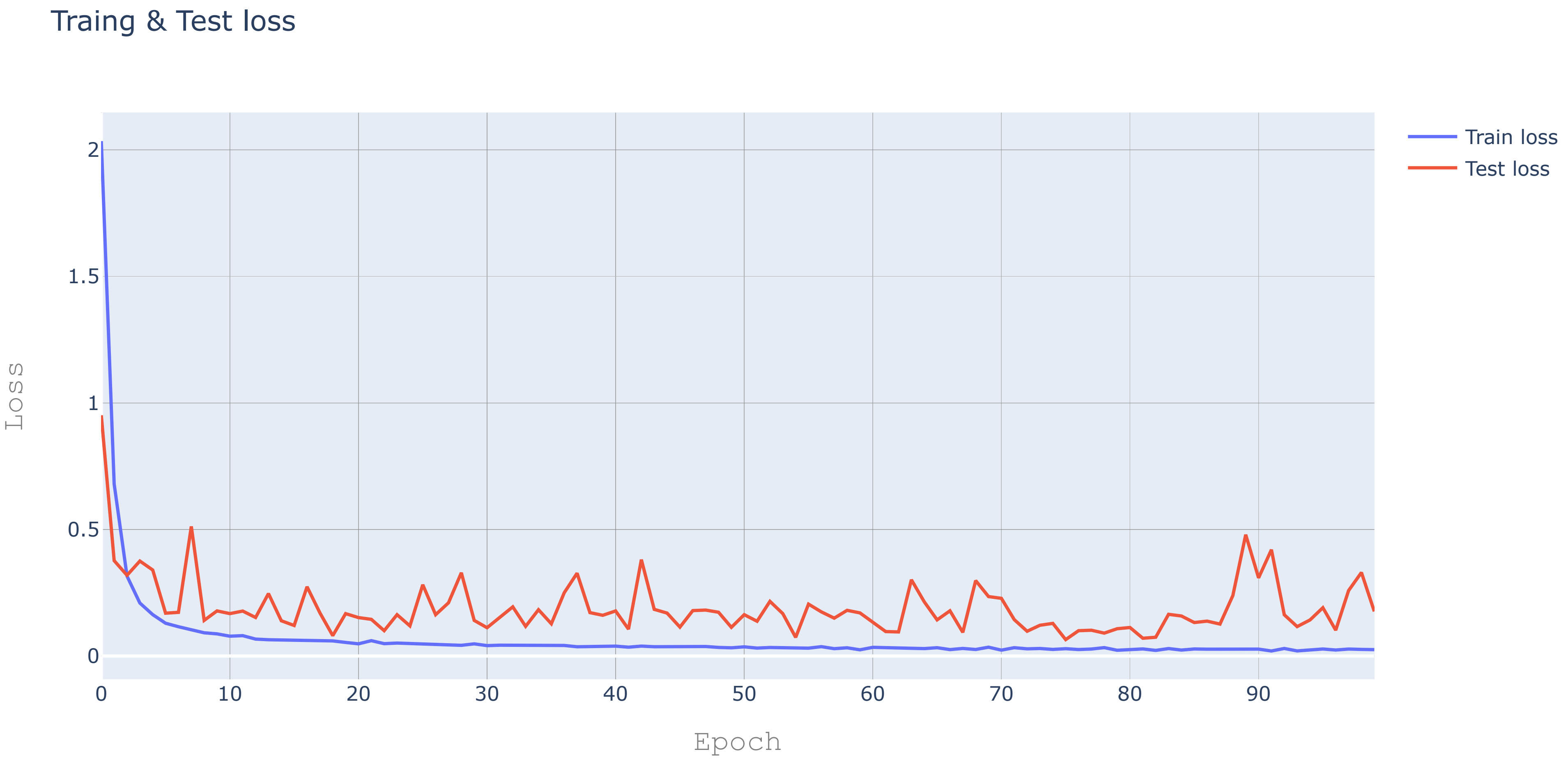
```
***EPOCH SUMMARY*** Loss: 0.03696431964635849 Acc: 0.9933169484138489 | Test Loss: 0.1743566393852234 Test Acc0.9786913990974426


***EPOCH SUMMARY*** Loss: 0.02900967188179493 Acc: 0.9941588044166565 | Test Loss: 0.14955347776412964 Test Acc0.9829689860343933


***EPOCH SUMMARY*** Loss: 0.032504282891750336 Acc: 0.9932915568351746 | Test Loss: 0.18052184581756592 Test Acc0.9829688668251038


***EPOCH SUMMARY*** Loss: 0.024556146934628487 Acc: 0.9951280951499939 | Test Loss: 0.17033377289772034 Test Acc0.9826520681381226


***EPOCH SUMMARY*** Loss: 0.034279078245162964 Acc: 0.9940567016601562 | Test Loss: 0.13459159433841705 Test Acc0.9832066297531128


***EPOCH SUMMARY*** Loss: 0.03316066786646843 Acc: 0.9941588044166565 | Test Loss: 0.09664713591337204 Test Acc0.9857414364814758


***EPOCH SUMMARY*** Loss: 0.02815011702477932 Acc: 0.9945668578147888 | Test Loss: 0.09515415877103806 Test Acc0.9869296550750732


***EPOCH SUMMARY*** Loss: 0.027523569762706757 Acc: 0.9942863583564758 | Test Loss: 0.3013705313205719 Test Acc0.9733840227127075


***EPOCH SUMMARY*** Loss: 0.029309658333659172 Acc: 0.9943628907203674 | Test Loss: 0.21422944962978363 Test Acc0.9824144244194031


***EPOCH SUMMARY*** Loss: 0.03281288594007492 Acc: 0.9937251210212708 | Test Loss: 0.14281496405601501 Test Acc0.9858999252319336


***EPOCH SUMMARY*** Loss: 0.025397730991244316 Acc: 0.9951790571212769 | Test Loss: 0.1786300241947174 Test Acc0.9820184111595154


***EPOCH SUMMARY*** Loss: 0.029932191595435143 Acc: 0.9943627715110779 | Test Loss: 0.0934901088476181 Test Acc0.9889099597930908
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.025823814794421196 Acc: 0.994770884513855 | Test Loss: 0.2981313467025757 Test Acc0.9683143496513367


***EPOCH SUMMARY*** Loss: 0.034825392067432404 Acc: 0.9939547181129456 | Test Loss: 0.2348274290561676 Test Acc0.9826520681381226


***EPOCH SUMMARY*** Loss: 0.023205844685435295 Acc: 0.9955106377601624 | Test Loss: 0.22790494561195374 Test Acc0.9839195609092712


***EPOCH SUMMARY*** Loss: 0.03292728215456009 Acc: 0.9939547181129456 | Test Loss: 0.14376036822795868 Test Acc0.9820184111595154
```

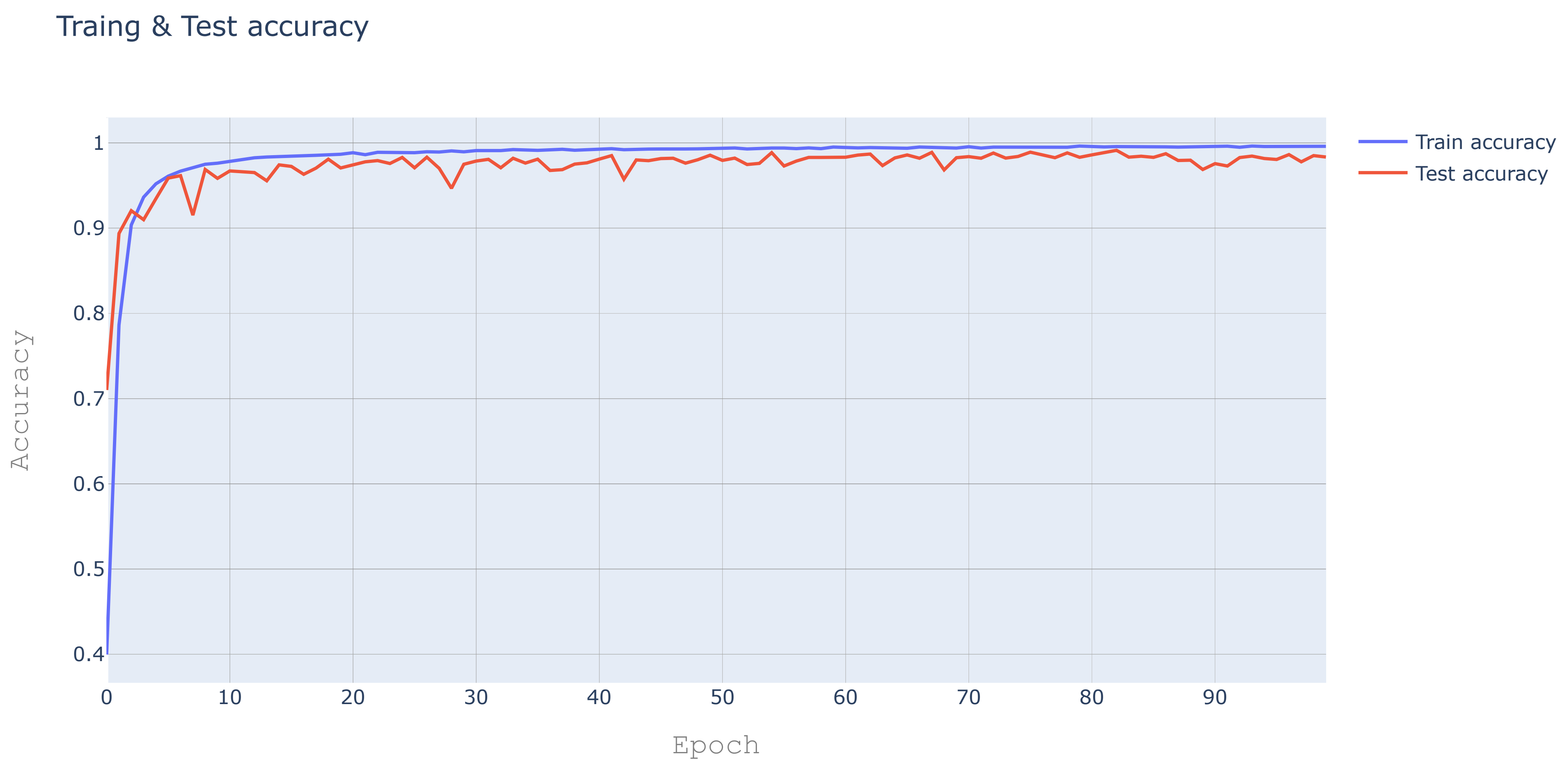
```
***EPOCH SUMMARY*** Loss: 0.02837499976158142 Acc: 0.9950515627861023 | Test Loss: 0.09787134826183319 Test Acc0.9881178736686707


***EPOCH SUMMARY*** Loss: 0.029736768454313278 Acc: 0.9948985576629639 | Test Loss: 0.12157728523015976 Test Acc0.9821767807006836


***EPOCH SUMMARY*** Loss: 0.026004433631896973 Acc: 0.9951280951499939 | Test Loss: 0.12891320884227753 Test Acc0.9840779304504395


***EPOCH SUMMARY*** Loss: 0.028962302953004837 Acc: 0.9948220252990723 | Test Loss: 0.06477038562297821 Test Acc0.9891476035118103
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.025618603453040123 Acc: 0.9948220252990723 | Test Loss: 0.10007119923830032 Test Acc0.9858205914497375


***EPOCH SUMMARY*** Loss: 0.02761119231581688 Acc: 0.9954086542129517 | Test Loss: 0.10188484936952591 Test Acc0.9826520681381226


***EPOCH SUMMARY*** Loss: 0.03314771130681038 Acc: 0.9949239492416382 | Test Loss: 0.09042999893426895 Test Acc0.9882763624191284


***EPOCH SUMMARY*** Loss: 0.022731125354766846 Acc: 0.9963014125823975 | Test Loss: 0.10777890682220459 Test Acc0.9831273555755615


***EPOCH SUMMARY*** Loss: 0.025303708389401436 Acc: 0.9955106973648071 | Test Loss: 0.1126658245921135 Test Acc0.9859790802001953


***EPOCH SUMMARY*** Loss: 0.02800130471587181 Acc: 0.9952811598777771 | Test Loss: 0.07024144381284714 Test Acc0.9885931611061096


***EPOCH SUMMARY*** Loss: 0.022218801081180573 Acc: 0.9956892132759094 | Test Loss: 0.07412517815828323 Test Acc0.9912864565849304
Model saved in path: ./model.ckpt


***EPOCH SUMMARY*** Loss: 0.029473677277565002 Acc: 0.9952555894851685 | Test Loss: 0.16473889350891113 Test Acc0.9832859039306641


***EPOCH SUMMARY*** Loss: 0.023510005325078964 Acc: 0.9952300786972046 | Test Loss: 0.15816742181777954 Test Acc0.9843948483467102


***EPOCH SUMMARY*** Loss: 0.027842938899993896 Acc: 0.9954341650009155 | Test Loss: 0.1323118656873703 Test Acc0.983048141002655


***EPOCH SUMMARY*** Loss: 0.026967491954565048 Acc: 0.9953575730323792 | Test Loss: 0.13784345984458923 Test Acc0.9873257875442505
```

```
***EPOCH SUMMARY*** Loss: 0.02755080536007881 Acc: 0.9951025247573853 | Test Loss: 0.12619808316230774 Test Acc0.9794043302536011


***EPOCH SUMMARY*** Loss: 0.023473426699638367 Acc: 0.9952811598777771 | Test Loss: 0.2389213740825653 Test Acc0.9797212481498718


***EPOCH SUMMARY*** Loss: 0.024631358683109283 Acc: 0.9958932995796204 | Test Loss: 0.47949278354644775 Test Acc0.9687896966934204


***EPOCH SUMMARY*** Loss: 0.027279246598482132 Acc: 0.9951790571212769 | Test Loss: 0.30874258279800415 Test Acc0.9756813049316406


***EPOCH SUMMARY*** Loss: 0.019858475774526596 Acc: 0.9961482882499695 | Test Loss: 0.42014357447624207 Test Acc0.9729087948799133


***EPOCH SUMMARY*** Loss: 0.029789315536618233 Acc: 0.994924008846283 | Test Loss: 0.16297529637813568 Test Acc0.9828106164932251


***EPOCH SUMMARY*** Loss: 0.020172597840428352 Acc: 0.9963269233703613 | Test Loss: 0.11623883247375488 Test Acc0.9845532178878784


***EPOCH SUMMARY*** Loss: 0.02583465166389942 Acc: 0.9957912564277649 | Test Loss: 0.1424514353275299 Test Acc0.9817014932632446


***EPOCH SUMMARY*** Loss: 0.027680709958076477 Acc: 0.9958168268203735 | Test Loss: 0.1909451186656952 Test Acc0.9805924296379089


***EPOCH SUMMARY*** Loss: 0.023712942376732826 Acc: 0.9957147240638733 | Test Loss: 0.10205227881669998 Test Acc0.9862959980964661


***EPOCH SUMMARY*** Loss: 0.027506504207849503 Acc: 0.9955361485481262 | Test Loss: 0.2586253583431244 Test Acc0.977820098400116


***EPOCH SUMMARY*** Loss: 0.026076653972268105 Acc: 0.9956381916999817 | Test Loss: 0.330061137676239 Test Acc0.9851077795028687


***EPOCH SUMMARY*** Loss: 0.025170600041747093 Acc: 0.9960207939147949 | Test Loss: 0.17633837461471558 Test Acc0.9834442734718323
```

In [15]:

```
titlefont = dict(family='Courier New, monospace', size=18, color='#7f7f7f')
layout = go.Layout(title='Traing & Test loss', xaxis=dict(title='Epoch', titlefont=titlefont), yaxis=dict(title='Loss', titlefont=titlefont))
fig = go.Figure(data=[go.Scatter(y=train_history['loss'], name='Train loss'), go.Scatter(y=train_history['val_loss'], name='Test loss')], layout=layout)
plotly.offline.iplot(fig)
layout = go.Layout(title='Traing & Test accuracy', xaxis=dict(title='Epoch', titlefont=titlefont),
yaxis=dict(title='Accuracy', titlefont=titlefont))
fig = go.Figure(data=[go.Scatter(y=train_history['acc'], name='Train accuracy'), go.Scatter(y=train_history['val_acc'], name='Test accuracy')], layout=layout)
plotly.offline.iplot(fig)
```


Traing & Test loss
Train loss
Test loss
Loss
Epoch
0
0.5
1
1.5
2
0
10
20
30
40
50
60
70
80
90



In [26]:

```
saver.restore(sess, "./model.ckpt")
sess.run(test_init_op)
confusion_matrix = np.zeros([43,43])
test_analys = trainDf.copy()
predictions = []
probabilities = []
analys = []
for i in tqdm.tqdm_notebook(range(samples_test//batch_size), ascii=True, desc='Test best model'):
    _files, _predictions, _probas, _gts, _cm = sess.run([load_filename, pred_classes, pred_probas, load_label, confusion_matrix_op], feed_dict={dp_rate: 0, manual_load: False, im
g_placeholder: zero_img, label_placeholder: zero_label})
    confusion_matrix += _cm
    for i in range(batch_size):
        sample_info = {'image': _files[i].decode(), 'prediction': int(_predictions[i]), 'gt': int(_gts[i]), 'gt_probas': _probas[i][_gts[i]],'prediction_probas': _probas[i][_pred
ictions[i]], 'prediction_type': 'Correct' if _gts[i] ==_predictions[i] else 'Wrong'}
        for cls_id, j in enumerate(_probas[i]):
            sample_info['prob_{}'.format(cls_id)] = j
        analys.append(sample_info)
analys_df = pd.DataFrame(analys)
```

```
INFO:tensorflow:Restoring parameters from ./model.ckpt

C:\Users\vkkre\anaconda3\lib\site-packages\ipykernel_launcher.py:8: TqdmDeprecationWarning:

This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
```

In [27]:

```
saver.restore(sess, "./model.ckpt")
sess.run(test_init_op)
confusion_matrix = np.zeros([43,43])
test_analys = trainDf.copy()
predictions = []
probabilities = []
analys = []
for i in tqdm.tqdm_notebook(range(samples_test//batch_size), ascii=True, desc='Test best model'):
    _files, _predictions, _probas, _gts, _cm = sess.run([load_filename, pred_classes, pred_probas, load_label, confusion_matrix_op], feed_dict={dp_rate: 0, manual_load: False, im
g_placeholder: zero_img, label_placeholder: zero_label})
    confusion_matrix += _cm
    for i in range(batch_size):
        sample_info = {'image': _files[i].decode(), 'prediction': int(_predictions[i]), 'gt': int(_gts[i]), 'gt_probas': _probas[i][_gts[i]],'prediction_probas': _probas[i][_pred
ictions[i]], 'prediction_type': 'Correct' if _gts[i] ==_predictions[i] else 'Wrong'}
        for cls_id, j in enumerate(_probas[i]):
            sample_info['prob_{}'.format(cls_id)] = j
        analys.append(sample_info)
analys_df = pd.DataFrame(analys)
```

```
INFO:tensorflow:Restoring parameters from ./model.ckpt

C:\Users\vkkre\anaconda3\lib\site-packages\ipykernel_launcher.py:8: TqdmDeprecationWarning:

This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
```

In [28]:

```
analys_df.sample(4)
```

Out[28]:

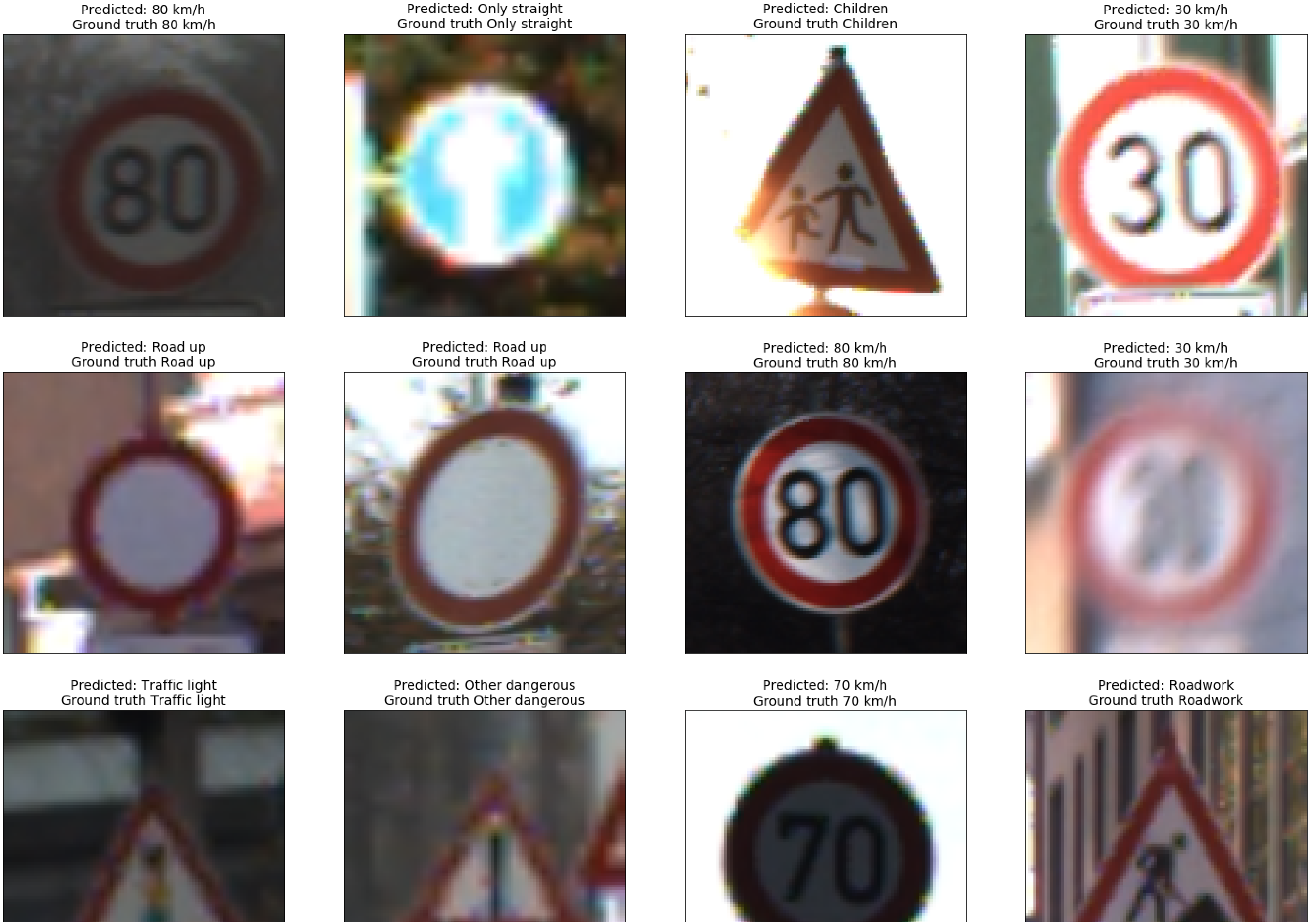
	image	prediction	gt	gt_probas	prediction_probas	prediction_type	prob_0	prob_1	prob_2	prob_3	...	prob_33	prob_34	prob_35	prob_36	prob_37	prob_
11843	./test/09250.png	13	13	1.000000	1.000000	Correct	1.450171e-30	0.000000	2.423994e-33	1.783680e-36	...	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000
6335	./test/01041.png	9	9	1.000000	1.000000	Correct	0.000000e+00	0.000000	0.000000e+00	0.000000e+00	...	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000
10342	./test/03872.png	25	25	1.000000	1.000000	Correct	0.000000e+00	0.000000	0.000000e+00	0.000000e+00	...	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000
11486	./test/10210.png	6	6	0.994619	0.994619	Correct	6.368614e-08	0.000317	4.950012e-07	7.422835e-06	...	4.112654e-08	2.761807e-11	9.355172e-13	2.810410e-10	2.040425e-12	0.000

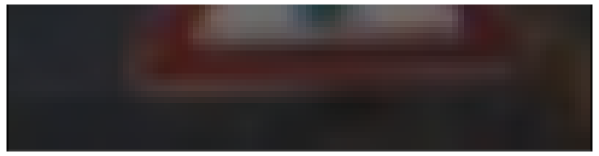
4 rows × 49 columns

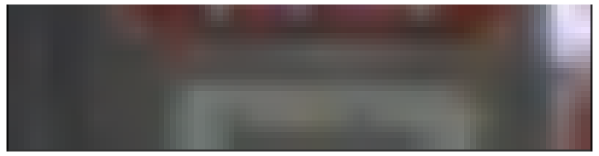
In [29]:

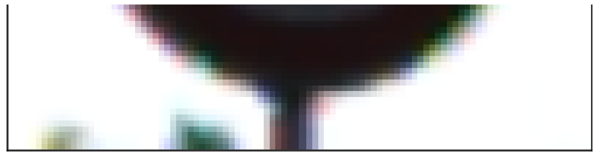
```
rows = 3
cols = 4
fig, axs = plt.subplots(rows, cols, sharex=True, sharey=True, figsize=(25, 8))
visualize = trainDf.sample(rows*cols)
analys_df_copy = analys_df.copy()
analys_df_copy = analys_df_copy.sample(frac=1)
idx = 0
for i in range(rows):
    for j in range(cols):
        img = cv2.imread(analys_df_copy.iloc[idx]['image'])
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (100, 100))
        gt = analys_df_copy.iloc[idx]['gt']
        pred = analys_df_copy.iloc[idx]['prediction']
        axs[i,j].imshow(img)
        axs[i,j].set_title('Predicted: {}\nGround truth {}'.format(labels[pred], labels[gt]),
        fontsize=14)
        axs[i,j].get_xaxis().set_visible(False)
        axs[i,j].get_yaxis().set_visible(False)
        idx += 1
fig.suptitle("Random prediction", fontsize=30, y=2.1, x=0.515);
plt.subplots_adjust(left=None, bottom=None, right=0.9, top=1.9, wspace=None, hspace=None)
```

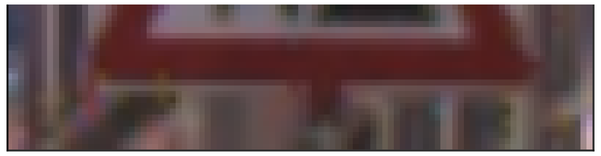
Random prediction





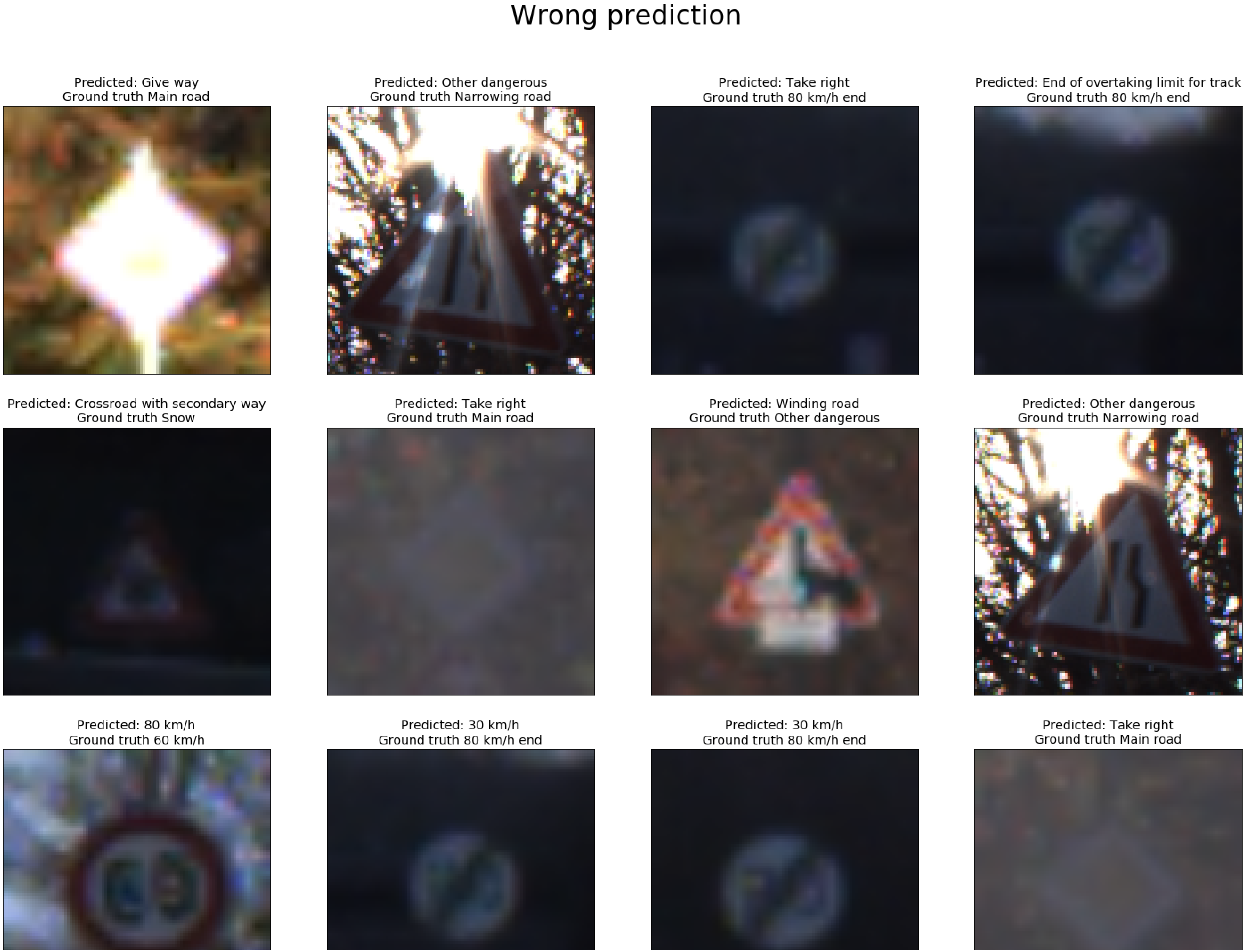


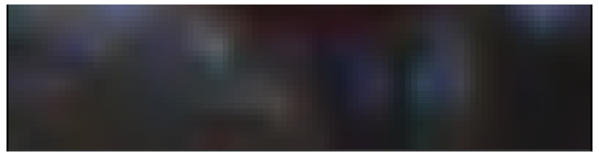


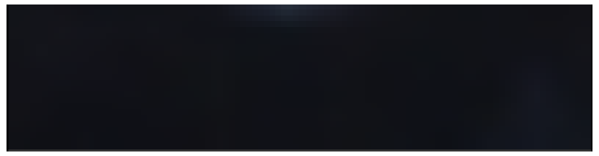


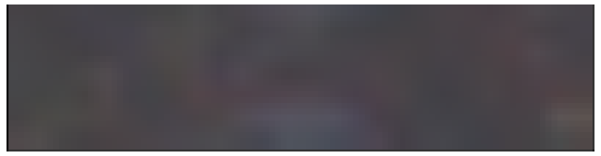
In [30]:

```
rows = 3
cols = 4
fig, axs = plt.subplots(rows, cols, sharex=True, sharey=True, figsize=(25, 8))
visualize = trainDf.sample(rows*cols)
analys_df_copy = analys_df[analys_df['prediction_type'] == 'Wrong'].copy()
analys_df_copy = analys_df_copy.sample(frac=1)
idx = 0
for i in range(rows):
    for j in range(cols):
        img = cv2.imread(analys_df_copy.iloc[idx]['image'])
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (100, 100))
        gt = analys_df_copy.iloc[idx]['gt']
        pred = analys_df_copy.iloc[idx]['prediction']
        axs[i,j].imshow(img)
        axs[i,j].set_title('Predicted: {}\nGround truth {}'.format(labels[pred], labels[gt]),
        fontsize=14)
        axs[i,j].get_xaxis().set_visible(False)
        axs[i,j].get_yaxis().set_visible(False)
        idx += 1
fig.suptitle("Wrong prediction", fontsize=30, y=2.1, x=0.515);
plt.subplots_adjust(left=None, bottom=None, right=0.9, top=1.9, wspace=None, hspace=None)
```


Wrong prediction
Predicted: Give way
Ground truth Main road
Predicted: Other dangerous
Ground truth Narrowing road
Predicted: Take right
Ground truth 80 km/h end
Predicted: End of overtaking limit for track
Ground truth 80 km/h end
Predicted: Crossroad with secondary way
Ground truth Snow
Predicted: Take right
Ground truth Main road
Predicted: Winding road
Ground truth Other dangerous
Predicted: Other dangerous
Ground truth Narrowing road
Predicted: 80 km/h
Ground truth 60 km/h
Predicted: 30 km/h
Ground truth 80 km/h end
Predicted: 30 km/h
Ground truth 80 km/h end
Predicted: Take right
Ground truth Main road

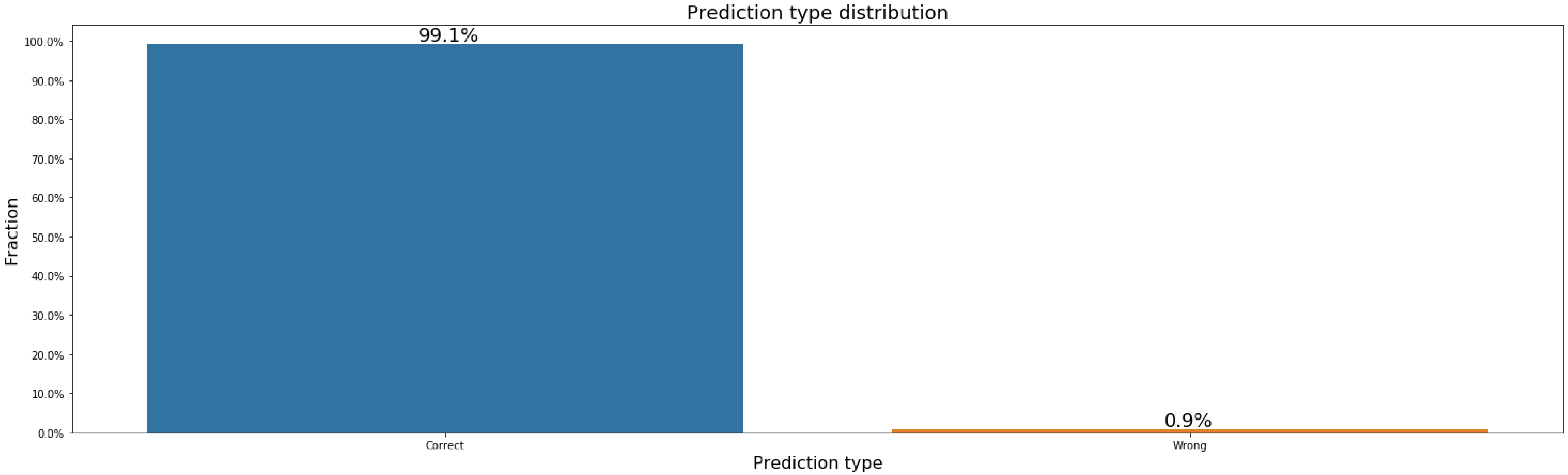






In [31]:

```
fig, axs = plt.subplots(1, 1, sharex=False, sharey=True, figsize=(25, 7))
px = sns.countplot(x='prediction_type', data=analys_df, ax=axs)
axs.set_title('Prediction type distribution', fontsize=18)
axs.set_xlabel('Prediction type', fontsize=16)
axs.set_ylabel('Fraction', fontsize=16);
total = analys_df.shape[0]
for idx, p in enumerate(px.patches):
    px.annotate('{:.1f}%'.format(p.get_height()/total*100), (p.get_x()+0.365,
    p.get_height()+100), fontsize=18)
    px.yaxis.set_ticks(np.linspace(0, total, 11))
    px.set_yticklabels(map('{:.1f}%'.format, 100*px.yaxis.get_majorticklocs()/total));
```



In [ ]: