Supplementary

Appendix 1

The principal codes of the PanycNet neural network model.

The principal codes of the PanycNet neural network model:

```
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.models import Sequential, load_model, Model
from tensorflow.keras.layers import Input, Conv2D, GlobalAveragePooling2D, BatchNormalization, Flatten
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras import layers, models, backend
def get_model(inputs, nb_classes):
# inputs = Input(shape=(31, 40, 1), name=“input”)
x = Conv2D(32, (1, 1), strides=1, activation=“relu”, name=“conv1”)(inputs)
# x = BatchNormalization()(x)
x = Conv2D(64, (3, 3), strides=2, name=“conv2”)(x)
x = Conv2D(32, (1, 1), strides=1, activation=“relu”, name=“conv3”)(x)
x = Flatten()(x)
# x = Dense(50, activation=“relu”, name=“dense1”)(x)
x = Dense(10, activation=“relu”, name=“dense2”)(x)
predictions = Dense(nb_classes, activation=“sigmoid”)(x)
# this is the model we will train
model = Model(inputs=inputs, outputs=predictions)
return model
def get_pre_model(inputs, nb_classes, weights=“imagenet”):
x = Conv2D(64, (3, 3), strides=1, activation=“relu”)(inputs)
x = layers.Conv2DTranspose(3, kernel_size=(4, 4),strides=(3, 3), padding=“same”, kernel_initializer=“he_normal”)(x)
base_model = InceptionV3(weights=weights, include_top=False)
x = base_model(x)
# add a global spatial average pooling layer
x = GlobalAveragePooling2D()(x)
# let’s add a fully-connected layer
# x = Dense(256, activation=“relu”)(x)
x = Dense(1024, activation=“relu”)(x)
# and a logistic layer
predictions = Dense(nb_classes, activation=“softmax”)(x)
# this is the model we will train
model = Model(inputs=inputs, outputs=predictions)
return model
```

https://dx.doi.org/10.21037/qims-23-970