Projet : Deep Food

Nous utilisons le GPU pour bénéficier de l'entraînement en précision mixte.

```
!nvidia-smi -L
```

```
GPU 0: NVIDIA GeForce RTX 2060 (UUID: GPU-a2b4317d-436b-2cc1-0fe3-ac16673637a5)
```

Obtenir des fonctions d'assistance.

```
# Importer une série de fonctions d'assistance pour le notebook
from helper_functions import create_tensorboard_callback, plot_loss_curves, compare_historys
```

Nous utilisons Utilisez TensorFlow datasets pour télécharger les données.

- Si vous voulez un aperçu, lisez le guide : https://www.tensorflow.org/datasets/overview?hl=fr
- Documentation de food101 sur TensorFlow : https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/

```
# Obtenir les datasets de TensorFlow
import tensorflow_datasets as tfds

# Lister tous les datasets disponibles
tfds_list = tfds.list_builders()
tfds_list
```

```
['abstract_reasoning',
 'accentdb',
 'aeslc',
 'aflw2k3d',
 'ag_news_subset',
 'ai2_arc',
 'ai2_arc_with_ir',
 'aloha_mobile',
 'amazon_us_reviews',
 'anli',
 'answer_equivalence',
 'arc',
 'asqa',
 'asset',
 'assin2',
 'asu_table_top_converted_externally_to_rlds',
 'austin_buds_dataset_converted_externally_to_rlds',
 'austin_sailor_dataset_converted_externally_to_rlds',
 'austin_sirius_dataset_converted_externally_to_rlds',
 'bair_robot_pushing_small',
 'bc_z',
 'bccd',
```

```
 'beans',
 'bee_dataset',
 'beir',
 'berkeley_autolab_ur5',
 'berkeley_cable_routing',
 'berkeley_fanuc_manipulation',
 'berkeley_gnm_cory_hall',
 'berkeley_gnm_recon',
 'berkeley_gnm_sac_son',
 'berkeley_mvp_converted_externally_to_rlds',
 'berkeley_rpt_converted_externally_to_rlds',
 'big_patent',
 'bigearthnet',
 'billsum',
 'binarized_mnist',
 'binary_alpha_digits',
 'ble_wind_field',
 'blimp',
 'booksum',
 'bool_q',
 'bot_adversarial_dialogue',
 'bridge',
 'bridge_data_msr',
 'bucc',
 'c4',
 'c4_wsrs',
 'caltech101',
 'caltech_birds2010',
 'caltech_birds2011',
 'cardiotox',
 'cars196',
 'cassava',
 'cats_vs_dogs',
 'celeb_a',
 'celeb_a_hq',
 'cfq',
 'cherry_blossoms',
 'chexpert',
 'cifar10',
 'cifar100',
 'cifar100_n',
 'cifar10_1',
 'cifar10_corrupted',
 'cifar10_h',
 'cifar10_n',
 'citrus_leaves',
 'cityscapes',
 'civil_comments',
 'clevr',
```

```
 'clic',
 'clinc_oos',
 'cmaterdb',
 'cmu_franka_exploration_dataset_converted_externally_to_rlds',
 'cmu_play_fusion',
 'cmu_stretch',
 'cnn_dailymail',
 'coco',
 'coco_captions',
 'coil100',
 'colorectal_histology',
 'colorectal_histology_large',
 'columbia_cairlab_pusht_real',
 'common_voice',
 'conll2002',
 'conll2003',
 'conq_hose_manipulation',
 'controlled_noisy_web_labels',
 'coqa',
 'corr2cause',
 'cos_e',
 'cosmos_qa',
 'covid19',
 'covid19sum',
 'crema_d',
 'criteo',
 'cs_restaurants',
 'curated_breast_imaging_ddsm',
 'cycle_gan',
 'd4rl_adroit_door',
 'd4rl_adroit_hammer',
 'd4rl_adroit_pen',
 'd4rl_adroit_relocate',
 'd4rl_antmaze',
 'd4rl_mujoco_ant',
 'd4rl_mujoco_halfcheetah',
 'd4rl_mujoco_hopper',
 'd4rl_mujoco_walker2d',
 'dart',
 'databricks_dolly',
 'davis',
 'deep1b',
 'deep_weeds',
 'definite_pronoun_resolution',
 'dementiabank',
 'diabetic_retinopathy_detection',
 'diamonds',
 'dices',
 'div2k',
```

```
 'dlr_edan_shared_control_converted_externally_to_rlds',
 'dlr_sara_grid_clamp_converted_externally_to_rlds',
 'dlr_sara_pour_converted_externally_to_rlds',
 'dmlab',
 'dobbe',
 'doc_nli',
 'dolphin_number_word',
 'domainnet',
 'downsampled_imagenet',
 'drop',
 'dsprites',
 'dtd',
 'duke_ultrasound',
 'e2e_cleaned',
 'efron_morris75',
 'emnist',
 'eraser_multi_rc',
 'esnli',
 'eth_agent_affordances',
 'eurosat',
 'fashion_mnist',
 'flic',
 'flores',
 'fmb',
 'food101',
 'forest_fires',
 'fractal20220817_data',
 'fuss',
 'gap',
 'geirhos_conflict_stimuli',
 'gem',
 'genomics_ood',
 'german_credit_numeric',
 'gigaword',
 'glove100_angular',
 'glue',
 'goemotions',
 'gov_report',
 'gpt3',
 'gref',
 'groove',
 'grounded_scan',
 'gsm8k',
 'gtzan',
 'gtzan_music_speech',
 'hellaswag',
 'higgs',
 'hillstrom',
 'horses_or_humans',
```

```
 'howell',
 'i_naturalist2017',
 'i_naturalist2018',
 'i_naturalist2021',
 'iamlab_cmu_pickup_insert_converted_externally_to_rlds',
 'imagenet2012',
 'imagenet2012_corrupted',
 'imagenet2012_fewshot',
 'imagenet2012_multilabel',
 'imagenet2012_real',
 'imagenet2012_subset',
 'imagenet_a',
 'imagenet_lt',
 'imagenet_pi',
 'imagenet_r',
 'imagenet_resized',
 'imagenet_sketch',
 'imagenet_v2',
 'imagenette',
 'imagewang',
 'imdb_reviews',
 'imperialcollege_sawyer_wrist_cam',
 'io_ai_tech',
 'irc_disentanglement',
 'iris',
 'istella',
 'jaco_play',
 'kaist_nonprehensile_converted_externally_to_rlds',
 'kddcup99',
 'kitti',
 'kmnist',
 'kuka',
 'laion400m',
 'lambada',
 'lfw',
 'librispeech',
 'librispeech_lm',
 'libritts',
 'ljspeech',
 'lm1b',
 'locomotion',
 'lost_and_found',
 'lsun',
 'lvis',
 'malaria',
 'maniskill_dataset_converted_externally_to_rlds',
 'math_dataset',
 'math_qa',
 'mctaco',
```

```
 'media_sum',
 'mimic_play',
 'mlqa',
 'mnist',
 'mnist_corrupted',
 'movie_lens',
 'movie_rationales',
 'movielens',
 'moving_mnist',
 'mrqa',
 'mslr_web',
 'mt_opt',
 'mtnt',
 'multi_news',
 'multi_nli',
 'multi_nli_mismatch',
 'natural_instructions',
 'natural_questions',
 'natural_questions_open',
 'newsroom',
 'nsynth',
 'nyu_depth_v2',
 'nyu_door_opening_surprising_effectiveness',
 'nyu_franka_play_dataset_converted_externally_to_rlds',
 'nyu_rot_dataset_converted_externally_to_rlds',
 'ogbg_molpcba',
 'omniglot',
 'open_images_challenge2019_detection',
 'open_images_v4',
 'openbookqa',
 'opinion_abstracts',
 'opinosis',
 'opus',
 'oxford_flowers102',
 'oxford_iiit_pet',
 'para_crawl',
 'pass',
 'patch_camelyon',
 'paws_wiki',
 'paws_x_wiki',
 'penguins',
 'pet_finder',
 'pg19',
 'piqa',
 'places365_small',
 'placesfull',
 'plant_leaves',
 'plant_village',
 'plantae_k',
```

```
 'plex_robosuite',
 'protein_net',
 'q_re_cc',
 'qa4mre',
 'qasc',
 'qm9',
 'quac',
 'quality',
 'quickdraw_bitmap',
 'race',
 'radon',
 'real_toxicity_prompts',
 'reddit',
 'reddit_disentanglement',
 'reddit_tifu',
 'ref_coco',
 'resisc45',
 'rlu_atari',
 'rlu_atari_checkpoints',
 'rlu_atari_checkpoints_ordered',
 'rlu_control_suite',
 'rlu_dmlab_explore_object_rewards_few',
 'rlu_dmlab_explore_object_rewards_many',
 'rlu_dmlab_rooms_select_nonmatching_object',
 'rlu_dmlab_rooms_watermaze',
 'rlu_dmlab_seekavoid_arena01',
 'rlu_locomotion',
 'rlu_rwrl',
 'robo_set',
 'robomimic_mg',
 'robomimic_mh',
 'robomimic_ph',
 'robonet',
 'robosuite_panda_pick_place_can',
 'roboturk',
 'rock_paper_scissors',
 'rock_you',
 's3o4d',
 'salient_span_wikipedia',
 'samsum',
 'savee',
 'scan',
 'scene_parse150',
 'schema_guided_dialogue',
 'sci_tail',
 'scicite',
 'scientific_papers',
 'scrolls',
 'segment_anything',
```

```
'sentiment140',
'shapes3d',
'sift1m',
'simpte',
'siscore',
'smallnorb',
'smartwatch_gestures',
'snli',
'so2sat',
'speech_commands',
'spoc_robot',
'spoken_digit',
'squad',
'squad_question_generation',
'stanford_dogs',
'stanford_hydra_dataset_converted_externally_to_rlds',
'stanford_kuka_multimodal_dataset_converted_externally_to_rlds',
'stanford_mask_vit_converted_externally_to_rlds',
'stanford_online_products',
'stanford_robocook_converted_externally_to_rlds',
'star_cfq',
'starcraft_video',
'stl10',
'story_cloze',
'summscreen',
'sun397',
'super_glue',
'svhn_cropped',
'symmetric_solids',
'taco_play',
'tao',
'tatoeba',
'ted_hrlr_translate',
'ted_multi_translate',
'tedlium',
'tf_flowers',
'the300w_lp',
'tidybot',
'tiny_shakespeare',
'titanic',
'tokyo_u_lsmo_converted_externally_to_rlds',
'toto',
'trec',
'trivia_qa',
'tydi_qa',
'uc_merced',
'ucf101',
'ucsd_kitchen_dataset_converted_externally_to_rlds',
'ucsd_pick_and_place_dataset_converted_externally_to_rlds',
```

```
'uiuc_d3field',
'unified_qa',
'universal_dependencies',
'unnatural_instructions',
'usc_cloth_sim_converted_externally_to_rlds',
'user_libri_audio',
'user_libri_text',
'utaustin_mutex',
'utokyo_pr2_opening_fridge_converted_externally_to_rlds',
'utokyo_pr2_tabletop_manipulation_converted_externally_to_rlds',
'utokyo_saytap_converted_externally_to_rlds',
'utokyo_xarm_bimanual_converted_externally_to_rlds',
'utokyo_xarm_pick_and_place_converted_externally_to_rlds',
'vctk',
'vima_converted_externally_to_rlds',
'viola',
'visual_domain_decathlon',
'voc',
'voxceleb',
'voxforge',
'waymo_open_dataset',
'web_graph',
'web_nlg',
'web_questions',
'webvid',
'wider_face',
'wiki40b',
'wiki_auto',
'wiki_bio',
'wiki_dialog',
'wiki_table_questions',
'wiki_table_text',
'wikiann',
'wikihow',
'wikipedia',
'wikipedia_toxicity_subtypes',
'wine_quality',
'winogrande',
'wit',
'wit_kaggle',
'wmt13_translate',
'wmt14_translate',
'wmt15_translate',
'wmt16_translate',
'wmt17_translate',
'wmt18_translate',
'wmt19_translate',
'wmt_t2t_translate',
'wmt_translate',
```

```
 'wordnet',
 'wsc273',
 'xnli',
 'xquad',
 'xsum',
 'xtreme_pawsx',
 'xtreme_pos',
 'xtreme_s',
 'xtreme_xnli',
 'yahoo_ltrc',
 'yelp_polarity_reviews',
 'yes_no',
 'youtube_vis']
```

```
# Vérifiez si food101 est dans la liste des datasets
print("food101" in tfds_list)
```

```
True
```

Chargement des donées (prend environ 5 - 6 minutes)

```
(train_data, test_data), ds_info = tfds.load(
    name="food101",
    split=["train", "validation"],
    shuffle_files=True,
    as_supervised=True, # Les données seront retournées sous forme de tuple (données, étiquette)
    with_info=True # Donne des métadonnées (ds_info)
)
```

Remarque : Si le dataset a déjà été téléchargé, le code précédent ne le téléchargera pas à nouveau.

Caractéristiques de food101

```
ds_info.features
```

```
FeaturesDict({
    'image': Image(shape=(None, None, 3), dtype=uint8),
    'label': ClassLabel(shape=(), dtype=int64, num_classes=101),
})
```

```
# Obtenir les noms des classes
class_names = ds_info.features["label"].names
# Affiche les 10 premiers aliments du dataset
class_names[:10]
```

```
['apple_pie',
 'baby_back_ribs',
 'baklava',
 'beef_carpaccio',
 'beef_tartare',
```

```
 'beet_salad',
 'beignets',
 'bibimbap',
 'bread_pudding',
 'breakfast_burrito']
```

Explorer le dataset Food

- Noms des classes
- Forme de notre entrée (tenseurs d'images)
- Le type de données de nos entrées
- À quoi ressemblent les étiquettes ? (sont-elles encodées en one-hot ou label encoded)
- Les étiquettes correspondent-elles aux noms des classes ?

Prendre un échantillon des données d'entraînement

```
train_one_simple = train_data.take(1)

train_one_simple # Les échantiollns sont au format format (image tesneur), (label)
```

```
<_TakeDataset element_spec=(TensorSpec(shape=(None, None, 3), dtype=tf.uint8, name=None), TensorSpec(shape=(), dtype=tf.int64, name=None))>
```

```
# Afficher des informations sur notre échantillon d'entraînement
for image, label in train_one_simple:
    print(f"""
    Image shape : {image.shape}
    Image datatype : {image.dtype}
    Target class : {label}
    Class names string form : {class_names[label.numpy()]}
    """)
```

```
    Image shape : (384, 512, 3)
    Image datatype : <dtype: 'uint8'>
    Target class : 70
    Class names string form : pad_thai
```

```
# À quoi ressemble notre tenseur d'image ?
image
```

```
<tf.Tensor: shape=(384, 512, 3), dtype=uint8, numpy=
array([[[230, 229, 183],
        [231, 230, 184],
        [232, 231, 183],
        ...,
        [232, 233, 202],
        [229, 234, 204],
```

```
        [229, 236, 205]],

       [[228, 227, 179],
        [227, 226, 178],
        [226, 226, 176],
        ...,
        [234, 235, 204],
        [232, 237, 207],
        [234, 241, 210]],

       [[229, 229, 175],
        [229, 229, 175],
        [227, 227, 175],
        ...,
        [233, 234, 202],
        [232, 237, 205],
        [235, 242, 209]],

       ...,

       [[119,  97,  58],
        [131, 109,  70],
        [131, 109,  70],
        ...,
        [239, 241, 220],
        [238, 240, 219],
        [238, 240, 219]],

       [[120,  98,  61],
        [133, 111,  74],
        [135, 113,  76],
        ...,
        [240, 242, 221],
        [240, 242, 221],
        [240, 242, 221]],

       [[136, 114,  77],
        [135, 113,  76],
        [129, 107,  70],
        ...,
        [243, 245, 224],
        [244, 246, 225],
        [245, 247, 226]]], dtype=uint8)>
```

```
# Quelles sont les valeurs minimales et maximales de l'image, censées être entre 0 et 255 ?
import tensorflow as tf
tf.reduce_min(image), tf.reduce_max(image)
```

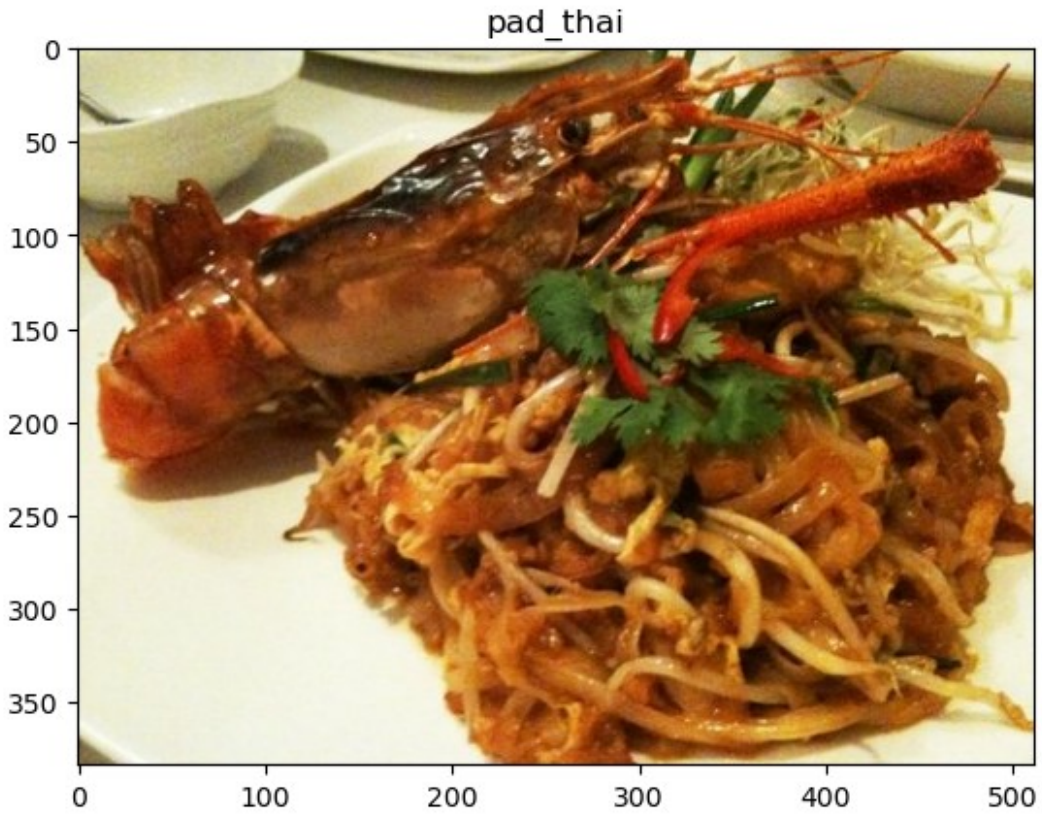
```
(<tf.Tensor: shape=(), dtype=uint8, numpy=0>,
 <tf.Tensor: shape=(), dtype=uint8, numpy=255>)
```

Neural netwroks like images to be encoded between 0 and 1

```
# Afficher une image du dataset TensorFlow
import matplotlib.pyplot as plt

plt.imshow(image)
plt.title(class_names[label.numpy()])
```

```
Text(0.5, 1.0, 'pad_thai')
```



Créer des fonctions de prétraitement pour nos données

Ce que nous savons sur nos données :

- Type de données 'uint8'
- Composé de tailles variées de tenseurs (images de tailles différentes)
- Non mises à l'échelle (les valeurs des pixels sont comprises entre 0 et 255)
- Ce que nous savons sur les modèles :

Données en 'float32'

- TensorFlow préfère que toutes les images d'un lot aient la même taille
- Mises à l'échelle (valeurs entre 0 et 1, également appelé normalisé)

- Avec cela en tête, nous avons plusieurs choses à aborder avec une fonction de prétraitement.

Comme nous allons utiliser EfficientNetBX pré-entraîné de tf.keras.applications, nous n'avons pas besoin de redimensionner nos données (ces architectures ont un redimensionnement intégré).

Notre fonction doit :

- Redimensionner toutes nos images à la même taille
- Convertir le type de données de nos tenseurs d'image de 'uint8' à 'float32'

```
# Créer une fonction pour le prétraitement des images
def preprocess_img(image, label, img_shape=244):
    """
    Convertir le type de données de l'image de 'uint8' à 'float32' et
redimensionner l'image à (img_shape, img_shape, 3).
    """
    image = tf.image.resize(image, [img_shape, img_shape])  # Reshape
target image
    return tf.cast(image, tf.float32), label

# Prétraiter une seule image échantillon et vérifier le résultat
preprocessed_img = preprocess_img(image, label)[0] # retourne un tuple
et nous voulons seulement l'image
print(f"Image before preprocessing:\n {image[:2]}")
print(f"Image shape: {image.shape}")
print(f"Image dtype: {image.dtype}")
print(f"############################")
print(f"Image after preprocessing:\n {preprocessed_img[:2]}")
print(f"Image shape: {preprocessed_img.shape}")
print(f"Image dtype: {preprocessed_img.dtype}")
```

```
Image before preprocessing:
 [[[230 229 183]
  [231 230 184]
  [232 231 183]
  ...
  [232 233 202]
  [229 234 204]
  [229 236 205]]

 [[228 227 179]
  [227 226 178]
  [226 226 176]
  ...
  [234 235 204]
  [232 237 207]
  [234 241 210]]]
Image shape: (384, 512, 3)
Image dtype: <dtype: 'uint8'>
############################
Image after preprocessing:
```

```
 [[[229.66031 228.66031 182.08653]
  [230.00269 229.28958 180.7158 ]
  [224.9142  224.9142  173.4224 ]
  ...
  [237.75494 231.47707 201.58101]
  [235.6259  233.38799 203.68315]
  [230.11931 236.02092 205.57011]]

 [[228.78413 228.64479 175.48085]
  [226.39359 226.39359 174.67227]
  [227.06    227.06    175.1308 ]
  ...
  [237.92068 232.17474 200.24554]
  [235.17233 233.49176 202.36893]
  [233.2896  239.19121 207.01909]]]
Image shape: (244, 244, 3)
Image dtype: <dtype: 'float32'>
```

Batcher et préparer les datasets

Nous allons maintenant rendre notre pipeline d'entrée de données vraiment rapide.

```
# Mapper la fonction de prétraitement sur les données d'entraînement
(AUTOTUNE utilise autant de processus que possible)
# Prétraiter et batcher les données d'entraînement
train_data = train_data.map(lambda x, y: preprocess_img(x, y,
img_shape=224), num_parallel_calls=tf.data.AUTOTUNE)
train_data =
train_data.shuffle(buffer_size=1000).batch(batch_size=32).prefetch(tf.
data.AUTOTUNE)

# Prétraiter et batcher les données de test
test_data = test_data.map(lambda x, y: preprocess_img(x, y,
img_shape=224), num_parallel_calls=tf.data.AUTOTUNE)
test_data = test_data.batch(batch_size=32).prefetch(tf.data.AUTOTUNE)

train_data, test_data
```

```
(<_PrefetchDataset element_spec=(TensorSpec(shape=(None, 224, 224, 3),
dtype=tf.float32, name=None), TensorSpec(shape=(None,),
dtype=tf.int64, name=None))>,
 <_PrefetchDataset element_spec=(TensorSpec(shape=(None, 224, 224, 3),
dtype=tf.float32, name=None), TensorSpec(shape=(None,),
dtype=tf.int64, name=None))>)
```

Créer des callbacks pour le modèle

Nous allons créer plusieurs callbacks pour nous aider pendant l'entraînement du modèle :

- Callback TensorBoard pour enregistrer les résultats de l'entraînement (pour que nous puissions les visualiser plus tard si nécessaire)

- Callback ModelCheckpoint pour sauvegarder la progression de notre modèle après l'extraction des caractéristiques

```
# Créer le callback TensorBoard (importé depuis helper_functions.py)
from helper_functions import create_tensorboard_callback

# Sauvegarder les poids au lieu de sauvegarder tout le modèle, ce qui est plus rapide
checkpoint_path = "model_checkpoints/cp.weights.h5"
model_checkpoint = tf.keras.callbacks.ModelCheckpoint(checkpoint_path,


monitor="val_acc",

save_best_only=True,

save_weights_only=True,
                                                      verbose=0)
# verbose = 0, ne rien afficher si le modèle est sauvegardé ou non
```

Configurer l'entraînement en précision mixte

```
# Activer l'entraînement en précision mixte
from tensorflow.keras import mixed_precision
mixed_precision.set_global_policy("mixed_float16") # Définir la politique de données globale en précision mixte
# Utiliser les types de données float16 et float32 pour accélérer les performances du modèle

mixed_precision.global_policy()
```

```
<FloatDTypePolicy "mixed_float16">
```

Construire un modèle d'extraction de caractéristiques

```
len(class_names)
```

```
101
```

```
from tensorflow.keras import layers

# Créer le modèle de base
input_shape = (224, 224, 3)
base_model = tf.keras.applications.EfficientNetV2B0(include_top=False)
# Geler les poids des paramètres du modèle de base
base_model.trainable = False

# Créer un modèle fonctionnel
inputs = layers.Input(shape=input_shape, name="input_layer")

x = base_model(inputs, training=False)
x = layers.GlobalAveragePooling2D()(x)
```

```
x = layers.Dense(len(class_names))(x)
outputs = layers.Activation("softmax", dtype=tf.float32,
name="softmax_float32")(x)

model = tf.keras.Model(inputs, outputs)

# Compiler le modèle
model.compile(loss="sparse_categorical_crossentropy",
optimizer=tf.keras.optimizers.Adam(), metrics=["accuracy"])

model.summary()
```

```
Model: "functional_1"
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━
┓
┃ Layer (type)                         ┃ Output Shape                ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━
┩
│ input_layer (InputLayer)             │ (None, 224, 224, 3)         │
0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────
┤
│ efficientnetv2-b0 (Functional)       │ (None, 7, 7, 1280)          │
5,919,312 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────
┤
│ global_average_pooling2d             │ (None, 1280)                │
0 │
│ (GlobalAveragePooling2D)             │                             │
│
├──────────────────────────────────────┼─────────────────────────────┼─────────────────
┤
│ dense (Dense)                        │ (None, 101)                 │
129,381 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────
┤
│ softmax_float32 (Activation)         │ (None, 101)                 │
0 │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────
┘
```

```
 Total params: 6,048,693 (23.07 MB)

 Trainable params: 129,381 (505.39 KB)

 Non-trainable params: 5,919,312 (22.58 MB)
```

Vérifier les politiques dtype des couches : utilisons-nous la précision mixte ?

```
# Vérifier les attributs dtype_policy des couches de notre modèle
for layer in model.layers:
    print(layer.name, layer.trainable, layer.dtype,
layer.dtype_policy)
```

```
input_layer True float32 <FloatDTypePolicy "mixed_float16">
efficientnetv2-b0 False float32 <FloatDTypePolicy "mixed_float16">
global_average_pooling2d True float32 <FloatDTypePolicy
"mixed_float16">
dense True float32 <FloatDTypePolicy "mixed_float16">
softmax_float32 True float32 <FloatDTypePolicy "float32">
```

```
# Vérifier les attributs dtype_policy des couches du modèle
EfficientV2NetB0
for layer in base_model.layers:
    print(layer.name, layer.trainable, layer.dtype,
layer.dtype_policy)
```

```
input_layer False float32 <FloatDTypePolicy "mixed_float16">
rescaling False float32 <FloatDTypePolicy "mixed_float16">
normalization False float32 <FloatDTypePolicy "mixed_float16">
stem_conv False float32 <FloatDTypePolicy "mixed_float16">
stem_bn False float32 <FloatDTypePolicy "mixed_float16">
stem_activation False float32 <FloatDTypePolicy "mixed_float16">
block1a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block1a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block1a_project_activation False float32 <FloatDTypePolicy
"mixed_float16">
block2a_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block2a_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block2a_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block2a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block2a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block2b_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block2b_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block2b_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block2b_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block2b_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block2b_drop False float32 <FloatDTypePolicy "mixed_float16">
block2b_add False float32 <FloatDTypePolicy "mixed_float16">
block3a_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block3a_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block3a_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block3a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block3a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block3b_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
```

```
block3b_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block3b_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block3b_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block3b_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block3b_drop False float32 <FloatDTypePolicy "mixed_float16">
block3b_add False float32 <FloatDTypePolicy "mixed_float16">
block4a_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block4a_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block4a_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block4a_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block4a_bn False float32 <FloatDTypePolicy "mixed_float16">
block4a_activation False float32 <FloatDTypePolicy "mixed_float16">
block4a_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block4a_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block4a_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block4a_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block4a_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block4a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block4a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block4b_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block4b_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block4b_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block4b_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block4b_bn False float32 <FloatDTypePolicy "mixed_float16">
block4b_activation False float32 <FloatDTypePolicy "mixed_float16">
block4b_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block4b_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block4b_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block4b_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block4b_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block4b_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block4b_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block4b_drop False float32 <FloatDTypePolicy "mixed_float16">
block4b_add False float32 <FloatDTypePolicy "mixed_float16">
block4c_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block4c_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block4c_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block4c_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block4c_bn False float32 <FloatDTypePolicy "mixed_float16">
block4c_activation False float32 <FloatDTypePolicy "mixed_float16">
block4c_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block4c_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block4c_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block4c_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block4c_se_excite False float32 <FloatDTypePolicy "mixed_float16">
```

```
block4c_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block4c_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block4c_drop False float32 <FloatDTypePolicy "mixed_float16">
block4c_add False float32 <FloatDTypePolicy "mixed_float16">
block5a_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block5a_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block5a_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block5a_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block5a_bn False float32 <FloatDTypePolicy "mixed_float16">
block5a_activation False float32 <FloatDTypePolicy "mixed_float16">
block5a_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block5a_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block5a_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block5a_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block5a_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block5a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block5a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block5b_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block5b_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block5b_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block5b_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block5b_bn False float32 <FloatDTypePolicy "mixed_float16">
block5b_activation False float32 <FloatDTypePolicy "mixed_float16">
block5b_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block5b_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block5b_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block5b_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block5b_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block5b_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block5b_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block5b_drop False float32 <FloatDTypePolicy "mixed_float16">
block5b_add False float32 <FloatDTypePolicy "mixed_float16">
block5c_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block5c_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block5c_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block5c_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block5c_bn False float32 <FloatDTypePolicy "mixed_float16">
block5c_activation False float32 <FloatDTypePolicy "mixed_float16">
block5c_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block5c_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block5c_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block5c_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block5c_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block5c_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block5c_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block5c_drop False float32 <FloatDTypePolicy "mixed_float16">
```

```
block5c_add False float32 <FloatDTypePolicy "mixed_float16">
block5d_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block5d_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block5d_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block5d_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block5d_bn False float32 <FloatDTypePolicy "mixed_float16">
block5d_activation False float32 <FloatDTypePolicy "mixed_float16">
block5d_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block5d_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block5d_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block5d_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block5d_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block5d_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block5d_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block5d_drop False float32 <FloatDTypePolicy "mixed_float16">
block5d_add False float32 <FloatDTypePolicy "mixed_float16">
block5e_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block5e_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block5e_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block5e_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block5e_bn False float32 <FloatDTypePolicy "mixed_float16">
block5e_activation False float32 <FloatDTypePolicy "mixed_float16">
block5e_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block5e_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block5e_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block5e_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block5e_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block5e_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block5e_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block5e_drop False float32 <FloatDTypePolicy "mixed_float16">
block5e_add False float32 <FloatDTypePolicy "mixed_float16">
block6a_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6a_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block6a_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block6a_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6a_bn False float32 <FloatDTypePolicy "mixed_float16">
block6a_activation False float32 <FloatDTypePolicy "mixed_float16">
block6a_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6a_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6a_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6a_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6a_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6b_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6b_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
```

```
block6b_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block6b_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6b_bn False float32 <FloatDTypePolicy "mixed_float16">
block6b_activation False float32 <FloatDTypePolicy "mixed_float16">
block6b_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6b_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6b_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6b_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6b_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6b_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6b_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6b_drop False float32 <FloatDTypePolicy "mixed_float16">
block6b_add False float32 <FloatDTypePolicy "mixed_float16">
block6c_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6c_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block6c_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block6c_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6c_bn False float32 <FloatDTypePolicy "mixed_float16">
block6c_activation False float32 <FloatDTypePolicy "mixed_float16">
block6c_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6c_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6c_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6c_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6c_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6c_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6c_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6c_drop False float32 <FloatDTypePolicy "mixed_float16">
block6c_add False float32 <FloatDTypePolicy "mixed_float16">
block6d_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6d_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block6d_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block6d_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6d_bn False float32 <FloatDTypePolicy "mixed_float16">
block6d_activation False float32 <FloatDTypePolicy "mixed_float16">
block6d_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6d_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6d_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6d_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6d_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6d_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6d_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6d_drop False float32 <FloatDTypePolicy "mixed_float16">
block6d_add False float32 <FloatDTypePolicy "mixed_float16">
block6e_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6e_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block6e_expand_activation False float32 <FloatDTypePolicy
```

"mixed_float16">
block6e_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6e_bn False float32 <FloatDTypePolicy "mixed_float16">
block6e_activation False float32 <FloatDTypePolicy "mixed_float16">
block6e_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6e_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6e_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6e_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6e_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6e_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6e_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6e_drop False float32 <FloatDTypePolicy "mixed_float16">
block6e_add False float32 <FloatDTypePolicy "mixed_float16">
block6f_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6f_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block6f_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block6f_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6f_bn False float32 <FloatDTypePolicy "mixed_float16">
block6f_activation False float32 <FloatDTypePolicy "mixed_float16">
block6f_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6f_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6f_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6f_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6f_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6f_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6f_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6f_drop False float32 <FloatDTypePolicy "mixed_float16">
block6f_add False float32 <FloatDTypePolicy "mixed_float16">
block6g_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6g_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block6g_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block6g_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6g_bn False float32 <FloatDTypePolicy "mixed_float16">
block6g_activation False float32 <FloatDTypePolicy "mixed_float16">
block6g_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6g_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6g_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6g_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6g_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6g_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6g_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6g_drop False float32 <FloatDTypePolicy "mixed_float16">
block6g_add False float32 <FloatDTypePolicy "mixed_float16">
block6h_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block6h_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block6h_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">

```
block6h_dwconv2 False float32 <FloatDTypePolicy "mixed_float16">
block6h_bn False float32 <FloatDTypePolicy "mixed_float16">
block6h_activation False float32 <FloatDTypePolicy "mixed_float16">
block6h_se_squeeze False float32 <FloatDTypePolicy "mixed_float16">
block6h_se_reshape False float32 <FloatDTypePolicy "mixed_float16">
block6h_se_reduce False float32 <FloatDTypePolicy "mixed_float16">
block6h_se_expand False float32 <FloatDTypePolicy "mixed_float16">
block6h_se_excite False float32 <FloatDTypePolicy "mixed_float16">
block6h_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block6h_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block6h_drop False float32 <FloatDTypePolicy "mixed_float16">
block6h_add False float32 <FloatDTypePolicy "mixed_float16">
top_conv False float32 <FloatDTypePolicy "mixed_float16">
top_bn False float32 <FloatDTypePolicy "mixed_float16">
top_activation False float32 <FloatDTypePolicy "mixed_float16">

mixed_precision.global_policy()

<FloatDTypePolicy "mixed_float16">
```

Entraîner le modèle d'extraction de caractéristiques

```
# Entraîner le modèle d'extraction de caractéristiques avec des
callbacks
history_101_food_classes_feature_extract = model.fit(train_data,
                                                     epochs=3,

steps_per_epoch=len(train_data),

validation_data=test_data,

validation_steps=int(0.15 * len(test_data)),

callbacks=[create_tensorboard_callback(dir_name="training_logs",

experiment_name="efficientnetb0_101_classes_all_data_feature_extract")
,

model_checkpoint])

Saving TensorBoard log files to:
training_logs/efficientnetb0_101_classes_all_data_feature_extract/
20241003-090308
Epoch 1/3
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 3742s 2s/step - accuracy: 0.4600 -
loss: 2.3543 - val_accuracy: 0.6962 - val_loss: 1.1668
Epoch 2/3

C:\Users\anasa\AppData\Roaming\jupyterlab-desktop\jlab_server\Lib\
site-packages\keras\src\callbacks\model_checkpoint.py:206:
UserWarning: Can save best model only with val_acc available,
```

```
skipping.
  self._save_model(epoch=epoch, batch=None, logs=logs)
C:\Users\anasa\AppData\Roaming\jupyterlab-desktop\jlab_server\Lib\
contextlib.py:158: UserWarning: Your input ran out of data;
interrupting training. Make sure that your dataset or generator can
generate at least `steps_per_epoch * epochs` batches. You may need to
use the `.repeat()` function when building your dataset.
  self.gen.throw(value)

2368/2368 ━━━━━━━━━━━━━━━━━━━━ 168s 71ms/step - accuracy: 0.0000e+00 -
loss: 0.0000e+00 - val_accuracy: 0.6841 - val_loss: 1.1820
Epoch 3/3
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 3672s 2s/step - accuracy: 0.6586 -
loss: 1.3329 - val_accuracy: 0.7142 - val_loss: 1.0480
```

```
# Évaluer le modèle sur l'ensemble du dataset de test
results_feature_extract_model = model.evaluate(test_data)
results_feature_extract_model
```

```
790/790 ━━━━━━━━━━━━━━━━━━━━ 1159s 1s/step - accuracy: 0.7168 - loss:
1.0524

[1.0519026517868042, 0.7169108986854553]
```

```
# Sauvegarder le modèle localement (si vous utilisez Google Colab, le
modèle sauvegardé sera supprimé lorsque l'instance Colab sera
terminée)
save_dir =
"07_efficientnetb0_feature_extract_model_mixed_precision.keras"
model.save(save_dir)

# Charger le modèle précédemment sauvegardé
loaded_saved_model = tf.keras.models.load_model(save_dir)

# Vérifier les couches du modèle de base et voir quelle politique
dtype elles utilisent
for layer in loaded_saved_model.layers[1].layers[:20]: # vérifier
uniquement les 20 premières couches pour limiter la sortie
    print(layer.name, layer.trainable, layer.dtype,
layer.dtype_policy)
```

```
input_layer True float32 <FloatDTypePolicy "mixed_float16">
rescaling False float32 <FloatDTypePolicy "mixed_float16">
normalization False float32 <FloatDTypePolicy "mixed_float16">
stem_conv False float32 <FloatDTypePolicy "mixed_float16">
stem_bn False float32 <FloatDTypePolicy "mixed_float16">
stem_activation False float32 <FloatDTypePolicy "mixed_float16">
block1a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block1a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block1a_project_activation False float32 <FloatDTypePolicy
"mixed_float16">
```

```
block2a_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block2a_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block2a_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block2a_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block2a_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block2b_expand_conv False float32 <FloatDTypePolicy "mixed_float16">
block2b_expand_bn False float32 <FloatDTypePolicy "mixed_float16">
block2b_expand_activation False float32 <FloatDTypePolicy
"mixed_float16">
block2b_project_conv False float32 <FloatDTypePolicy "mixed_float16">
block2b_project_bn False float32 <FloatDTypePolicy "mixed_float16">
block2b_drop False float32 <FloatDTypePolicy "mixed_float16">

# Vérifier les performances du modèle chargé (cela devrait être les
mêmes que celles du modèle d'extraction de caractéristiques)
results_loaded_saved_model = loaded_saved_model.evaluate(test_data)
results_loaded_saved_model

790/790 ━━━━━━━━━━━━━━━━━━━━ 1169s 1s/step - accuracy: 0.7166 - loss:
1.0528

[1.0519040822982788, 0.7169108986854553]

# Les résultats du modèle chargé doivent être égaux (ou du moins très
proches) de ceux du modèle avant la sauvegarde
# Remarque : cela ne fonctionnera que si vous avez instancié des
variables de résultats
# import numpy as np
# assert np.isclose(results_feature_extract_model,
results_loaded_saved_model, rtol=1e-05, atol=1e-08).all()

# Obtenir un résumé de notre modèle téléchargé
loaded_saved_model.summary()

Model: "functional_1"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━
┓
┃ Layer (type)                         ┃ Output Shape                ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━
┩
│ input_layer (InputLayer)             │ (None, 224, 224, 3)         │
0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────
┤
│ efficientnetv2-b0 (Functional)       │ (None, 7, 7, 1280)          │
5,919,312 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────
┤
```

```
| global_average_pooling2d         | (None, 1280)           |
0 |
| (GlobalAveragePooling2D)         |                        |
|
├──────────────────────────────────┼────────────────────────┼──────────
──────┤
| dense (Dense)                    | (None, 101)            |
129,381 |
├──────────────────────────────────┼────────────────────────┼──────────
──────┤
| softmax_float32 (Activation)     | (None, 101)            |
0 |
└──────────────────────────────────┴────────────────────────┴──────────
──────┘

 Total params: 6,307,461 (24.06 MB)

 Trainable params: 129,381 (505.39 KB)

 Non-trainable params: 5,919,312 (22.58 MB)

 Optimizer params: 258,768 (1010.82 KB)
```

```python
# Configurer le callback EarlyStopping pour arrêter l'entraînement si
la val_loss du modèle n'améliore pas pendant 3 époques
early_stopping = tf.keras.callbacks.EarlyStopping(monitor="val_loss",
# Surveiller la métrique val_loss
                                                  patience=3)  # Si la
val_loss diminue pendant 3 époques d'affilée, arrêter l'entraînement

# Créer un callback ModelCheckpoint pour sauvegarder le meilleur
modèle pendant le fine-tuning
checkpoint_path = "fine_tune_checkpoints/fine_tune_checkpoints.keras"
model_checkpoint = tf.keras.callbacks.ModelCheckpoint(checkpoint_path,

save_best_only=True,

monitor="val_loss")

# Créer un callback de réduction du taux d'apprentissage
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor="val_loss",
                                                 factor=0.2,  #
multiply the learning rate by 0.2 (reduce by 5x)
                                                 patience=2,
                                                 verbose=1,  # print
out when learning rate goes down
                                                 min_lr=1e-7)

# Compiler le modèle
loaded_saved_model.compile(loss="sparse_categorical_crossentropy",  #
sparse_categorical_crossentropy car les labels ne sont pas one-hot
```

```
                        optimizer=tf.keras.optimizers.Adam(0.0001),  #
10x moins que le taux d'apprentissage par défaut
                        metrics=["accuracy"])

# Commencer le fine-tuning (pour toutes les couches)
history_101_food_classes_all_data_fine_tune =
loaded_saved_model.fit(train_data,
                                                    epochs=100, #
Affiner pour un maximum de 100 époques

steps_per_epoch=len(train_data),

validation_data=test_data,

validation_steps=int(0.15 * len(test_data)), # Validation pendant
l'entraînement sur 15% des données de test

callbacks=[create_tensorboard_callback("training_logs",
"efficientb0_101_classes_all_data_fine_tuning"), # Garder trace des
logs du modèle pendant l'entraînement

model_checkpoint, # Sauvegarder uniquement le meilleur modèle pendant
l'entraînement

early_stopping, # Arrêter le modèle après X époques sans améliorations

reduce_lr]) # Réduire le taux d'apprentissage après X époques sans
améliorations

Saving TensorBoard log files to:
training_logs/efficientb0_101_classes_all_data_fine_tuning/20241003-
114824
Epoch 1/100
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 3656s 2s/step - accuracy: 0.7011 -
loss: 1.1496 - val_accuracy: 0.7381 - val_loss: 1.0137 -
learning_rate: 1.0000e-04
Epoch 2/100
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 167s 71ms/step - accuracy: 0.0000e+00 -
loss: 0.0000e+00 - val_accuracy: 0.7256 - val_loss: 1.0239 -
learning_rate: 1.0000e-04
Epoch 3/100
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 3800s 2s/step - accuracy: 0.7087 -
loss: 1.1279 - val_accuracy: 0.7336 - val_loss: 0.9921 -
learning_rate: 1.0000e-04
Epoch 4/100
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 181s 76ms/step - accuracy: 0.0000e+00 -
loss: 0.0000e+00 - val_accuracy: 0.7370 - val_loss: 0.9848 -
learning_rate: 1.0000e-04
Epoch 5/100
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 3849s 2s/step - accuracy: 0.7141 -
```

```
loss: 1.1101 - val_accuracy: 0.7264 - val_loss: 0.9995 -
learning_rate: 1.0000e-04
Epoch 6/100

Epoch 6: ReduceLROnPlateau reducing learning rate to
1.9999999494757503e-05.
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 196s 83ms/step - accuracy: 0.0000e+00 -
loss: 0.0000e+00 - val_accuracy: 0.7301 - val_loss: 1.0164 -
learning_rate: 1.0000e-04
Epoch 7/100
2368/2368 ━━━━━━━━━━━━━━━━━━━━ 3689s 2s/step - accuracy: 0.7208 -
loss: 1.0872 - val_accuracy: 0.7263 - val_loss: 1.0203 -
learning_rate: 2.0000e-05
```

```python
# Sauvegarder le modèle localement
loaded_saved_model.save("07_efficientnetb0_fine_tuned_101_classes_mixe
d_precision.keras")

# Charger le modèle fine-tuné depuis Google Storage et évaluer
loaded_fine_tuned_gs_model =
tf.keras.models.load_model("07_efficientnetb0_fine_tuned_101_classes_m
ixed_precision.keras")

# Obtenir un résumé du modèle (même architecture que ci-dessus)
loaded_fine_tuned_gs_model.summary()
```

```
Model: "functional_1"
```

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
efficientnetv2-b0 (Functional)	(None, 7, 7, 1280)	5,919,312
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dense (Dense)	(None, 101)	129,381

```
├─────────────────────────────────────┼─────────────────────────────┼──────────
──────┤
│ softmax_float32 (Activation)        │ (None, 101)                 │
0 │
└─────────────────────────────────────┴─────────────────────────────┴──────────
──────┘

 Total params: 6,307,461 (24.06 MB)

 Trainable params: 129,381 (505.39 KB)

 Non-trainable params: 5,919,312 (22.58 MB)

 Optimizer params: 258,768 (1010.82 KB)

results_downloaded_fine_tuned_gs_model =
loaded_fine_tuned_gs_model.evaluate(test_data)
results_downloaded_fine_tuned_gs_model

790/790 ━━━━━━━━━━━━━━━━━━━━ 1214s 2s/step - accuracy: 0.7362 - loss:
0.9983

[0.9974470138549805, 0.7340593934059143]
```