Importing Libraries

In [35]:
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img, img_
from tensorflow.keras.preprocessing import image
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import ConfusionMatrixDisplay
import os
import matplotlib.pyplot as plt
from PIL import Image
import numpy as np
import pandas as pd
import cv2
```

Setting up directory paths

In [2]:
```
# Base directory
base_dir = '.'

print(f"Base directory: {base_dir}")
```

```
Base directory: .
```

In [3]:
```
# Training data directory
train_dir = os.path.join(base_dir, 'train')

print(f"Training directory: {train_dir}")
```

```
Training directory: .\train
```

In [4]:
```
# Validation data directory
validation_dir = os.path.join(base_dir, 'validation')

print(f"Validation directory: {validation_dir}")
```

```
Validation directory: .\validation
```

In [5]:
```
# Sample images directory
sample_dir = os.path.join(base_dir, 'sample_images')

print(f"Sample images directory: {sample_dir}")
```

```
Sample images directory: .\sample_images
```

In [6]:
```
# How many images is in each folder
train_neozep = len(os.listdir(os.path.join(train_dir, 'Neozep')))
train_medicol = len(os.listdir(os.path.join(train_dir, 'Medicol')))
train_kremils = len(os.listdir(os.path.join(train_dir, 'KremilS')))
train_fishoil = len(os.listdir(os.path.join(train_dir, 'FishOil')))
train_decolgen = len(os.listdir(os.path.join(train_dir, 'Decolgen')))
train_dayzinc = len(os.listdir(os.path.join(train_dir, 'DayZinc')))
```

```
train_biogesic = len(os.listdir(os.path.join(train_dir, 'Biogesic')))
train_bioflu = len(os.listdir(os.path.join(train_dir, 'Bioflu')))
train_bactidol = len(os.listdir(os.path.join(train_dir, 'Bactidol')))
train_alaxan = len(os.listdir(os.path.join(train_dir, 'Alaxan')))

print(f"Training Neozep: {train_neozep} images")
print(f"Training Medicol: {train_medicol} images")
print(f"Training Kremil S: {train_kremils} images")
print(f"Training Fish Oil: {train_fishoil} images")
print(f"Training Decolgen: {train_decolgen} images")
print(f"Training DayZinc: {train_dayzinc} images")
print(f"Training Biogesic: {train_biogesic} images")
print(f"Training Bioflu: {train_bioflu} images")
print(f"Training Bactidol: {train_bactidol} images")
print(f"Training Alaxan: {train_alaxan} images")
```

```
Training Neozep: 800 images
Training Medicol: 800 images
Training Kremil S: 800 images
Training Fish Oil: 800 images
Training Decolgen: 800 images
Training DayZinc: 800 images
Training Biogesic: 800 images
Training Bioflu: 800 images
Training Bactidol: 800 images
Training Alaxan: 800 images
```

In [7]:

```
# How many images is in each folder
val_neozep = len(os.listdir(os.path.join(validation_dir, 'Neozep')))
val_medicol = len(os.listdir(os.path.join(validation_dir, 'Medicol')))
val_kremils = len(os.listdir(os.path.join(validation_dir, 'KremilS')))
val_fishoil = len(os.listdir(os.path.join(validation_dir, 'FishOil')))
val_decolgen = len(os.listdir(os.path.join(validation_dir, 'Decolgen')))
val_dayzinc = len(os.listdir(os.path.join(validation_dir, 'DayZinc')))
val_biogesic = len(os.listdir(os.path.join(validation_dir, 'Biogesic')))
val_bioflu = len(os.listdir(os.path.join(validation_dir, 'Bioflu')))
val_bactidol = len(os.listdir(os.path.join(validation_dir, 'Bactidol')))
val_alaxan = len(os.listdir(os.path.join(validation_dir, 'Alaxan')))

print(f"Validation Neozep: {val_neozep} images")
print(f"Validation Medicol: {val_medicol} images")
print(f"Validation Kremil S: {val_kremils} images")
print(f"Validation Fish Oil: {val_fishoil} images")
print(f"Validation Decolgen: {val_decolgen} images")
print(f"Validation DayZinc: {val_dayzinc} images")
print(f"Validation Biogesic: {val_biogesic} images")
print(f"Validation Bioflu: {val_bioflu} images")
print(f"Validation Bactidol: {val_bactidol} images")
print(f"Validation Alaxan: {val_alaxan} images")
```

```
Validation Neozep: 200 images
Validation Medicol: 200 images
Validation Kremil S: 200 images
Validation Fish Oil: 200 images
Validation Decolgen: 200 images
Validation DayZinc: 200 images
Validation Biogesic: 200 images
Validation Bioflu: 200 images
Validation Bactidol: 200 images
Validation Alaxan: 200 images
```

Sample image visualization

In [8]:
```
samp_image_path = os.path.join(sample_dir, '00000000.jpg')

print(f"Loading image from: {samp_image_path}")
```

```
Loading image from: .\sample_images\00000000.jpg
```

In [9]:
```
samp_image = Image.open(samp_image_path)

print(f"Image loaded successfully!")
print(f"Image format: {samp_image.format}")
print(f"Image mode: {samp_image.mode}")
print(f"Image size: {samp_image.size}")
```

```
Image loaded successfully!
Image format: JPEG
Image mode: RGB
Image size: (300, 300)
```

In [10]:
```
# display the image
plt.imshow(samp_image)
plt.axis('off')
plt.show()
```

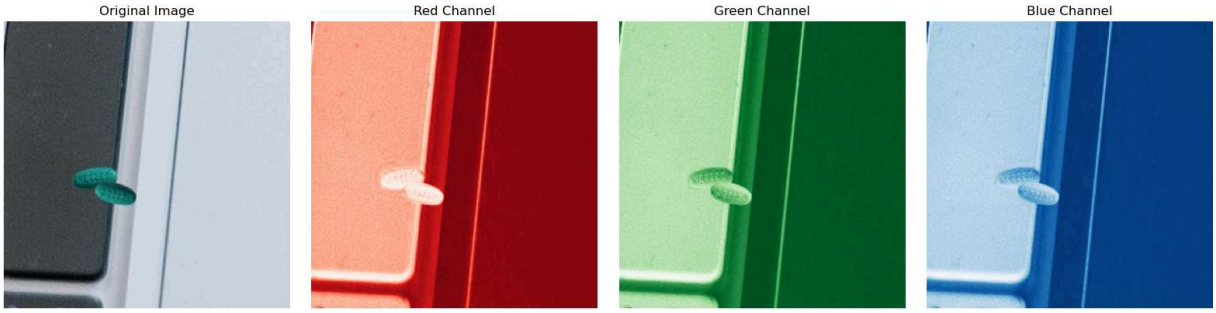
In [11]:
```
samp_array = np.array(samp_image)
```

In [12]:
```
# extract each color channel
red_channel = samp_array[:, :, 0]
green_channel = samp_array[:, :, 1]
blue_channel = samp_array[:, :, 2]

print(f"Red channel shape: {red_channel.shape}")
print(f"Green channel shape: {green_channel.shape}")
print(f"Blue channel shape: {blue_channel.shape}")
```

```
Red channel shape: (300, 300)
Green channel shape: (300, 300)
Blue channel shape: (300, 300)
```

In [13]:
```
# display all three channels side by side
fig, axes = plt.subplots(1, 4, figsize=(16, 4))

# original image
axes[0].imshow(samp_image)
axes[0].set_title('Original Image')
axes[0].axis('off')

# red channel
axes[1].imshow(red_channel, cmap='Reds')
axes[1].set_title('Red Channel')
axes[1].axis('off')

# green channel
axes[2].imshow(green_channel, cmap='Greens')
axes[2].set_title('Green Channel')
axes[2].axis('off')
```

```
# blue channel
axes[3].imshow(blue_channel, cmap='Blues')
axes[3].set_title('Blue Channel')
axes[3].axis('off')

plt.tight_layout()
plt.show()
```



In [14]:
```
# actual pixel values in a small 5x5 region
small_patch = samp_array[50:55, 50:55, 0]

print("Pixel values from a 5x5 patch (Red channel):")
print(small_patch)
```

```
Pixel values from a 5x5 patch (Red channel):
[[66 68 69 69 69]
 [66 69 70 69 69]
 [68 70 68 66 65]
 [72 71 66 61 59]
 [71 69 64 60 59]]
```

Sample image quality check

In [15]:
```
! pip install opencv-python
```

```
Collecting opencv-python
  Downloading opencv_python-4.12.0.88-cp37-abi3-win_amd64.whl.metadata (19 kB)
Collecting numpy<2.3.0,>=2 (from opencv-python)
  Downloading numpy-2.2.6-cp312-cp312-win_amd64.whl.metadata (60 kB)
Downloading opencv_python-4.12.0.88-cp37-abi3-win_amd64.whl (39.0 MB)
   ---------------------------------------- 0.0/39.0 MB ? eta -:--:--
   -- ------------------------------------- 2.4/39.0 MB 11.2 MB/s eta 0:00:04
   ---- ----------------------------------- 4.2/39.0 MB 10.1 MB/s eta 0:00:04
   ----- ---------------------------------- 5.5/39.0 MB 9.1 MB/s eta 0:00:04
   ------- -------------------------------- 7.1/39.0 MB 8.4 MB/s eta 0:00:04
   --------- ------------------------------ 9.4/39.0 MB 8.9 MB/s eta 0:00:04
   ------------ --------------------------- 11.8/39.0 MB 9.3 MB/s eta 0:00:03
   --------------- ------------------------ 15.2/39.0 MB 10.2 MB/s eta 0:00:03
   ------------------ --------------------- 18.1/39.0 MB 10.7 MB/s eta 0:00:02
   --------------------- ------------------ 21.0/39.0 MB 10.9 MB/s eta 0:00:02
   ------------------------ --------------- 23.9/39.0 MB 11.3 MB/s eta 0:00:02
   ------------------------- -------------- 27.0/39.0 MB 11.5 MB/s eta 0:00:02
   ------------------------------- -------- 30.4/39.0 MB 11.8 MB/s eta 0:00:01
   ---------------------------------- ----- 33.6/39.0 MB 12.1 MB/s eta 0:00:01
   ------------------------------------ --- 36.2/39.0 MB 12.1 MB/s eta 0:00:01
   ---------------------------------------  38.8/39.0 MB 12.1 MB/s eta 0:00:01
   ---------------------------------------- 39.0/39.0 MB 11.4 MB/s  0:00:03
Downloading numpy-2.2.6-cp312-cp312-win_amd64.whl (12.6 MB)
   ---------------------------------------- 0.0/12.6 MB ? eta -:--:--
   ------ --------------------------------- 2.1/12.6 MB 10.7 MB/s eta 0:00:01
   ----------- ---------------------------- 3.9/12.6 MB 9.0 MB/s eta 0:00:01
   ---------- ----------------------------- 6.8/12.6 MB 10.7 MB/s eta 0:00:01
   ------------------------------ --------- 9.7/12.6 MB 11.4 MB/s eta 0:00:01
   ---------------------------------------  12.6/12.6 MB 12.0 MB/s eta 0:00:01
   ---------------------------------------- 12.6/12.6 MB 11.3 MB/s  0:00:01
Installing collected packages: numpy, opencv-python

  Attempting uninstall: numpy

    Found existing installation: numpy 1.26.4

   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
    Uninstalling numpy-1.26.4:
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
```

```
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
   Successfully uninstalled numpy-1.26.4
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
```

```
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
---------------------------------------- 0/2 [numpy]
```

```
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------------------------------------- 0/2 [numpy]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------- ---------- 1/2 [opencv-python]
   ---------------------------------------- 2/2 [opencv-python]

Successfully installed numpy-2.2.6 opencv-python-4.12.0.88
  WARNING: Failed to remove contents in a temporary directory 'C:\Users\ASUS\anacond
a3\envs\tensorflow_env\Lib\site-packages\~umpy.libs'.
  You can safely remove it manually.
  WARNING: Failed to remove contents in a temporary directory 'C:\Users\ASUS\anacond
a3\envs\tensorflow_env\Lib\site-packages\~umpy'.
  You can safely remove it manually.
ERROR: pip's dependency resolver does not currently take into account all the packag
es that are installed. This behaviour is the source of the following dependency conf
licts.
tensorflow-intel 2.16.1 requires numpy<2.0.0,>=1.26.0; python_version >= "3.12", but
you have numpy 2.2.6 which is incompatible.
```

In [36]:

```python
# combination of two sources that I found online
# https://unimatrixz.com/blog/latent-space-measuring-image-quality-sharpness-clarit
# https://forum.opencv.org/t/scoring-the-clarity-of-an-image/16268
def image_quality(image_path):
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # sharpness
    laplacian = cv2.Laplacian(image, cv2.CV_64F)
    sharpness = laplacian.var()
```

```
    # sontrast
    contrast = image.std()

    # resolution (using Sobel operator)
    sobel_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=5)
    sobel_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=5)
    sobel = np.sqrt(sobel_x**2 + sobel_y**2)
    resolution = np.mean(sobel)

    threshold = 100

    if sharpness > threshold:
        print(f"The image is not blurry, Laplacian operator variance: {sharpness:.2
    else:
        print(f"The image is blurry, Laplacian operator variance: {sharpness:.2f}")

    return {
        "sharpness": sharpness.round(2),
        "resolution": resolution.round(2)
    }
```

In [37]:
```
image_quality(samp_image_path)
```

```
The image is not blurry, Laplacian operator variance: 123.91
```

Out[37]:
```
{'sharpness': 123.91, 'resolution': 330.23}
```

EDA visualization

In [17]:
```
names = [
    "Neozep", "Medicol", "Kremil S", "Fish Oil", "Decolgen","DayZinc", "Biogesic",
    "Bactidol", "Alaxan"
]

counts = [
    train_neozep, train_medicol, train_kremils, train_fishoil, train_decolgen, trai
    train_biogesic,train_bioflu, train_bactidol, train_alaxan
]

plt.bar(names, counts)
plt.xticks(rotation=45)
plt.xlabel("Medicine Names")
plt.ylabel("Number of Images")
plt.title("Training Set Class Distribution")
plt.tight_layout()
plt.show()
```

Creating ImageDataGenerator for training data

In [18]:
```
train_data = ImageDataGenerator(
    rescale = 1./255, # normalize pixels value to 0-1
    rotation_range = 40, # rotating images up 40 degrees
    width_shift_range = 0.2, # randomly shifting horizontally by 20%
    height_shift_range = 0.2,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True
)
```

In [19]:
```
val_data = ImageDataGenerator(
    rescale = 1./255
)
```

Create data generators from directories

In [20]:
```
target_size = (150,150)

train_generator = train_data.flow_from_directory(
    train_dir,
    target_size = target_size,
    batch_size = 32,
```

```
    class_mode = 'categorical'
)
```

```
Found 8000 images belonging to 10 classes.
```

In [21]:
```
val_generator = val_data.flow_from_directory(
    validation_dir,
    target_size = target_size,
    batch_size = 32,
    class_mode = 'categorical',
    shuffle = False
)
```

```
Found 2000 images belonging to 10 classes.
```

In [ ]:
```
t_lost,t_acc = model.evaluate(val_generator, verbose=1)
```

Visualize Data Augmentation

In [22]:
```
sample_path_img = os.path.join(sample_dir,'00000000.jpg')
sample_img = image.load_img(sample_path_img,target_size=target_size)
sample_img
```

Out[22]:

In [23]:
```
arr = image.img_to_array(sample_img)
arr_image = arr.reshape((1,)+arr.shape).shape
arr_image
```

Out[23]:
```
(1, 150, 150, 3)
```

In [24]:
```
samp_arr = image.img_to_array(sample_img)
# samp_arr.shape
samp_arr = samp_arr.reshape((1,)+(150, 150, 3))
samp_arr
samp_arr = image.img_to_array(sample_img)
# samp_arr.shape
samp_arr = samp_arr.reshape((1,)+(150, 150, 3))
samp_arr
```

Out[24]:
```
array([[[[195., 205., 217.],
         [ 43.,  53.,  62.],
         [ 36.,  45.,  50.],
         ...,
         [205., 215., 225.],
         [206., 216., 226.],
         [206., 216., 226.]],

        [[193., 203., 215.],
         [ 30.,  40.,  49.],
         [ 54.,  63.,  68.],
         ...,
         [205., 215., 225.],
         [205., 215., 225.],
         [205., 215., 225.]],

        [[193., 203., 215.],
         [ 21.,  31.,  40.],
         [ 63.,  72.,  77.],
         ...,
         [205., 215., 225.],
         [204., 214., 224.],
         [204., 214., 224.]],

        ...,

        [[ 59.,  66.,  74.],
         [ 66.,  73.,  81.],
         [ 59.,  66.,  74.],
         ...,
         [203., 213., 223.],
         [204., 214., 224.],
         [204., 214., 224.]],

        [[ 69.,  76.,  84.],
         [ 68.,  75.,  83.],
         [ 81.,  88.,  96.],
         ...,
         [201., 211., 221.],
         [203., 213., 223.],
         [202., 212., 222.]],

        [[ 68.,  75.,  83.],
         [ 67.,  74.,  82.],
         [ 90.,  97., 105.],
         ...,
         [201., 211., 221.],
         [203., 213., 223.],
         [200., 210., 220.]]]], dtype=float32)
```

In [25]:
```
# made another ImageDataGenerator specifically for the augmented images
# so that it does not rescale twice
aug_train_data = ImageDataGenerator(
    rotation_range = 40,
    width_shift_range = 0.2,
    height_shift_range = 0.2,
```

```
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True
)
```

In [38]:
```
# from butterfly note
def visualize_augmentation(generator, num_images=5):
    # get a batch of images
    x_batch, y_batch = next(generator)

    # take the first image
    img = x_batch[0]

    # create augmented versions
    fig, axes = plt.subplots(1, num_images, figsize=(15, 3))

    axes[0].imshow(img)
    axes[0].set_title('Original')
    axes[0].axis('off')

    # generate augmented versions
    img_array = img_to_array(img)
    img_array = img_array.reshape((1,) + img_array.shape)

    aug_iter = aug_train_data.flow(img_array, batch_size=1)

    for i in range(1, num_images):
        batch = next(aug_iter)
        aug_img = batch[0]
        axes[i].imshow(aug_img)
        axes[i].set_title(f'Augmented {i}')
        axes[i].axis('off')

    plt.tight_layout()
    plt.show()

visualize_augmentation(train_generator)
```

Building the model

In [27]:
```
model = Sequential([
    Conv2D(32, (3,3), activation='relu', padding='same', input_shape=(150,150,3)),
    BatchNormalization(),
    MaxPooling2D((2,2)),

    Conv2D(64, (3,3), activation='relu', padding='same'),
```

```
    BatchNormalization(),
    MaxPooling2D((2,2)),

    Conv2D(128, (3,3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((2,2)),

    Conv2D(256, (3,3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((2,2)),

    GlobalAveragePooling2D(),
    Dense(256, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(10, activation='softmax'),
])
```

```
C:\Users\ASUS\anaconda3\envs\tensorflow_env\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:113: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [28]:
```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	
conv2d (Conv2D)	(None, 150, 150, 32)	
batch_normalization (BatchNormalization)	(None, 150, 150, 32)	
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	
conv2d_1 (Conv2D)	(None, 75, 75, 64)	
batch_normalization_1 (BatchNormalization)	(None, 75, 75, 64)	
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	
conv2d_2 (Conv2D)	(None, 37, 37, 128)	
batch_normalization_2 (BatchNormalization)	(None, 37, 37, 128)	
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 128)	
conv2d_3 (Conv2D)	(None, 18, 18, 256)	
batch_normalization_3 (BatchNormalization)	(None, 18, 18, 256)	
max_pooling2d_3 (MaxPooling2D)	(None, 9, 9, 256)	
global_average_pooling2d (GlobalAveragePooling2D)	(None, 256)	
dense (Dense)	(None, 256)	
batch_normalization_4 (BatchNormalization)	(None, 256)	
dropout (Dropout)	(None, 256)	
dense_1 (Dense)	(None, 10)	

Total params: 459,722 (1.75 MB)

Trainable params: 458,250 (1.75 MB)

Non-trainable params: 1,472 (5.75 KB)

Compiling the model

In [29]:
```
model.compile(
    optimizer = 'adam',
    loss = 'categorical_crossentropy',
```

```
    metrics = ['accuracy'],
)
```

Train the model

In [30]:
```
early_stopping = EarlyStopping(
    monitor = 'val_loss',
    patience = 5,
    restore_best_weights = True,
)
```

In [29]:
```
history = model.fit(train_generator,epochs=20,validation_data=val_generator,callbac
```

```
Epoch 1/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 175s 683ms/step - accuracy: 0.1056 - loss: 2.9789 - val
_accuracy: 0.0990 - val_loss: 2.3707
Epoch 2/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 166s 665ms/step - accuracy: 0.1646 - loss: 2.4962 - val
_accuracy: 0.1060 - val_loss: 2.4087
Epoch 3/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 165s 660ms/step - accuracy: 0.3154 - loss: 1.9624 - val
_accuracy: 0.1880 - val_loss: 2.1716
Epoch 4/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 164s 654ms/step - accuracy: 0.5914 - loss: 1.1351 - val
_accuracy: 0.5420 - val_loss: 1.3239
Epoch 5/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 169s 674ms/step - accuracy: 0.7719 - loss: 0.6793 - val
_accuracy: 0.5795 - val_loss: 1.0886
Epoch 6/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 175s 699ms/step - accuracy: 0.8511 - loss: 0.4751 - val
_accuracy: 0.7910 - val_loss: 0.7050
Epoch 7/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 167s 667ms/step - accuracy: 0.8823 - loss: 0.3959 - val
_accuracy: 0.8675 - val_loss: 0.4474
Epoch 8/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 167s 668ms/step - accuracy: 0.8935 - loss: 0.3498 - val
_accuracy: 0.8310 - val_loss: 0.5268
Epoch 9/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 165s 658ms/step - accuracy: 0.9043 - loss: 0.3136 - val
_accuracy: 0.8260 - val_loss: 0.5933
Epoch 10/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 163s 650ms/step - accuracy: 0.9240 - loss: 0.2544 - val
_accuracy: 0.8675 - val_loss: 0.4428
Epoch 11/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 164s 657ms/step - accuracy: 0.9317 - loss: 0.2226 - val
_accuracy: 0.6480 - val_loss: 1.7206
Epoch 12/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 168s 672ms/step - accuracy: 0.9491 - loss: 0.1696 - val
_accuracy: 0.9330 - val_loss: 0.2423
Epoch 13/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 170s 678ms/step - accuracy: 0.9501 - loss: 0.1636 - val
_accuracy: 0.9295 - val_loss: 0.2124
Epoch 14/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 165s 658ms/step - accuracy: 0.9548 - loss: 0.1454 - val
_accuracy: 0.9235 - val_loss: 0.2403
Epoch 15/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 165s 659ms/step - accuracy: 0.9601 - loss: 0.1409 - val
_accuracy: 0.9310 - val_loss: 0.2988
Epoch 16/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 165s 661ms/step - accuracy: 0.9619 - loss: 0.1252 - val
_accuracy: 0.9380 - val_loss: 0.2226
Epoch 17/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 165s 658ms/step - accuracy: 0.9605 - loss: 0.1356 - val
_accuracy: 0.9240 - val_loss: 0.2609
Epoch 18/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 176s 702ms/step - accuracy: 0.9624 - loss: 0.1318 - val
_accuracy: 0.9630 - val_loss: 0.1358
Epoch 19/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 168s 670ms/step - accuracy: 0.9639 - loss: 0.1209 - val
```

```
_accuracy: 0.9505 - val_loss: 0.2011
Epoch 20/20
250/250 ━━━━━━━━━━━━━━━━━━━━ 167s 668ms/step - accuracy: 0.9690 - loss: 0.1130 - val
_accuracy: 0.9445 - val_loss: 0.1845
```

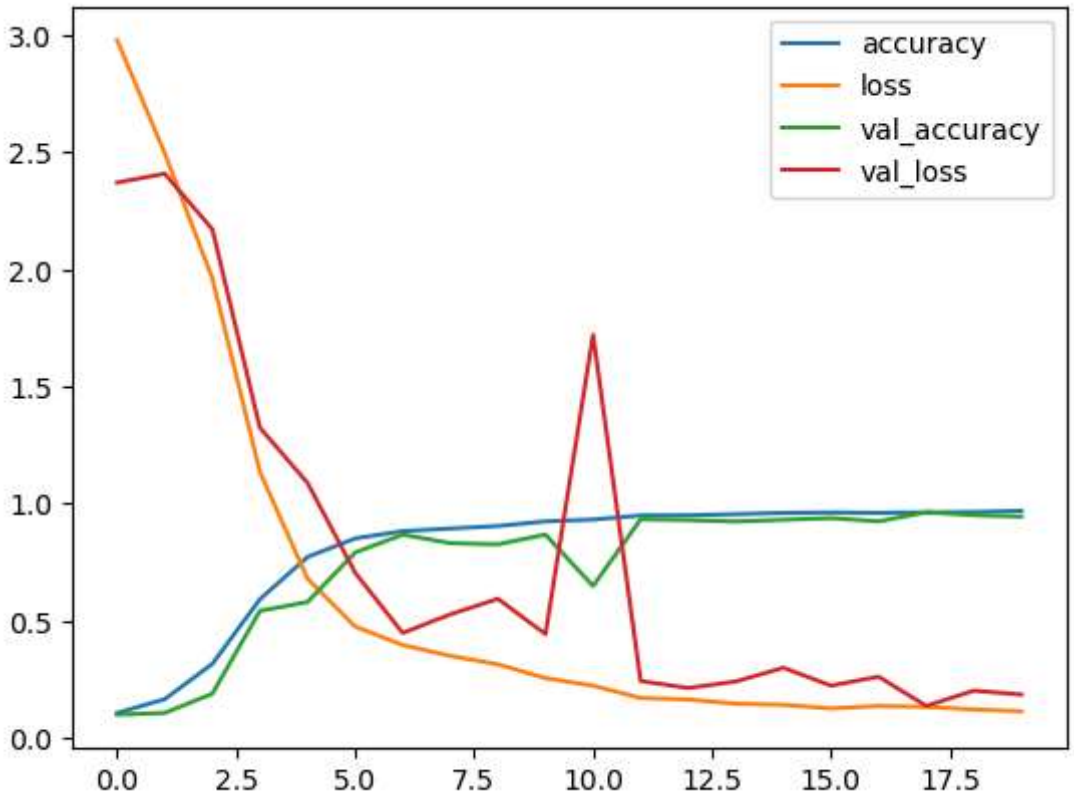
Save the trained model

In [30]:
```
saved_model = model.save('pill.keras')
```

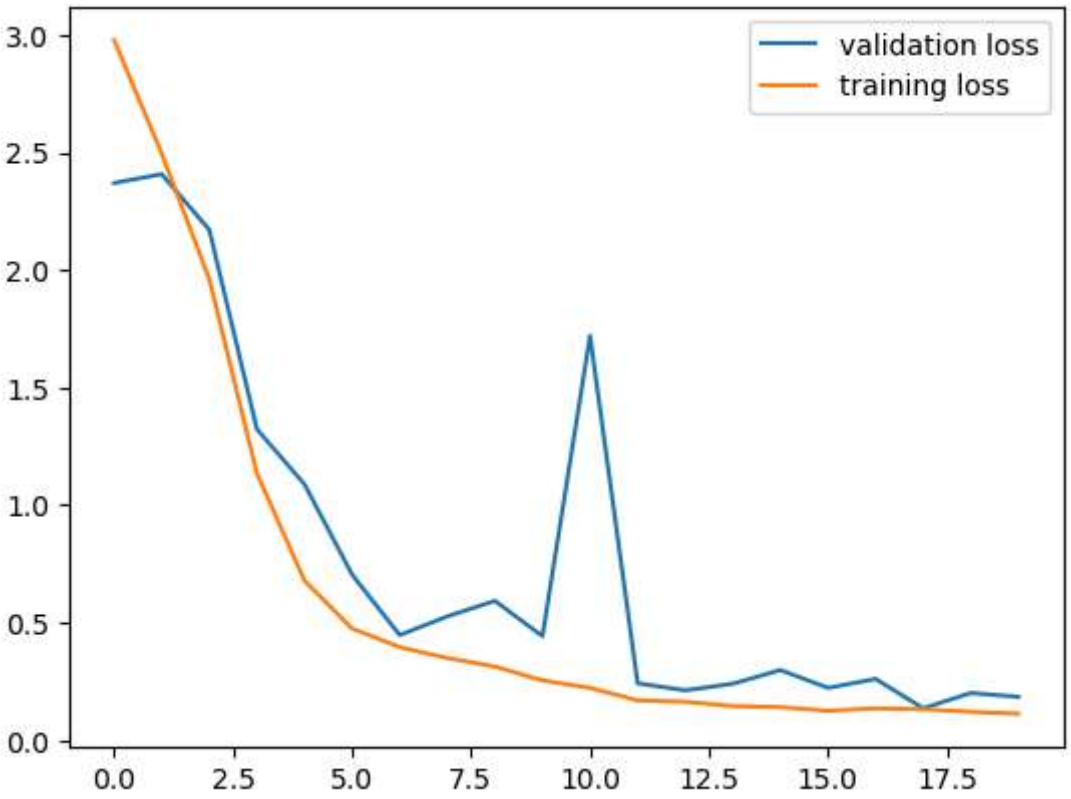
Evaluating the trained results

In [31]:
```
pd.DataFrame(history.history).plot()
```

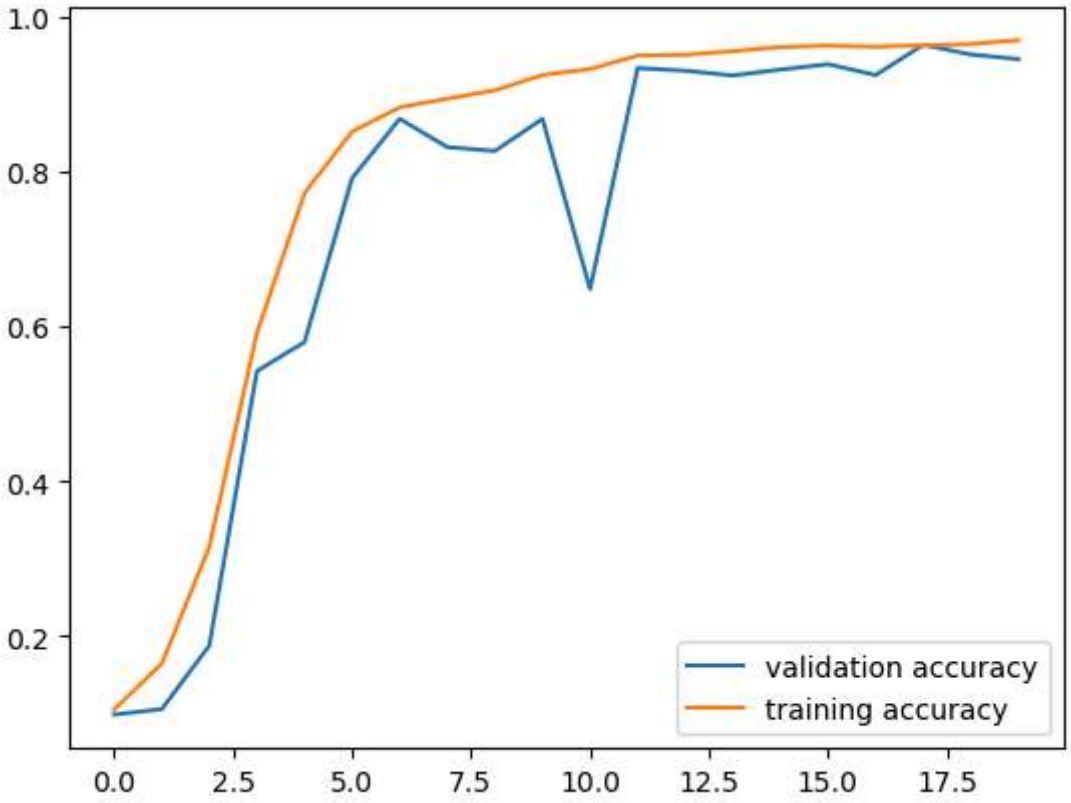
Out[31]: <Axes: >



In [32]:
```
plt.plot(history.history['val_loss'],label='validation loss')
plt.plot(history.history['loss'],label='training loss')
plt.legend()
plt.show()
```



In [33]:
```
plt.plot(history.history['val_accuracy'],label='validation accuracy')
plt.plot(history.history['accuracy'],label='training accuracy')
plt.legend()
plt.show()
```

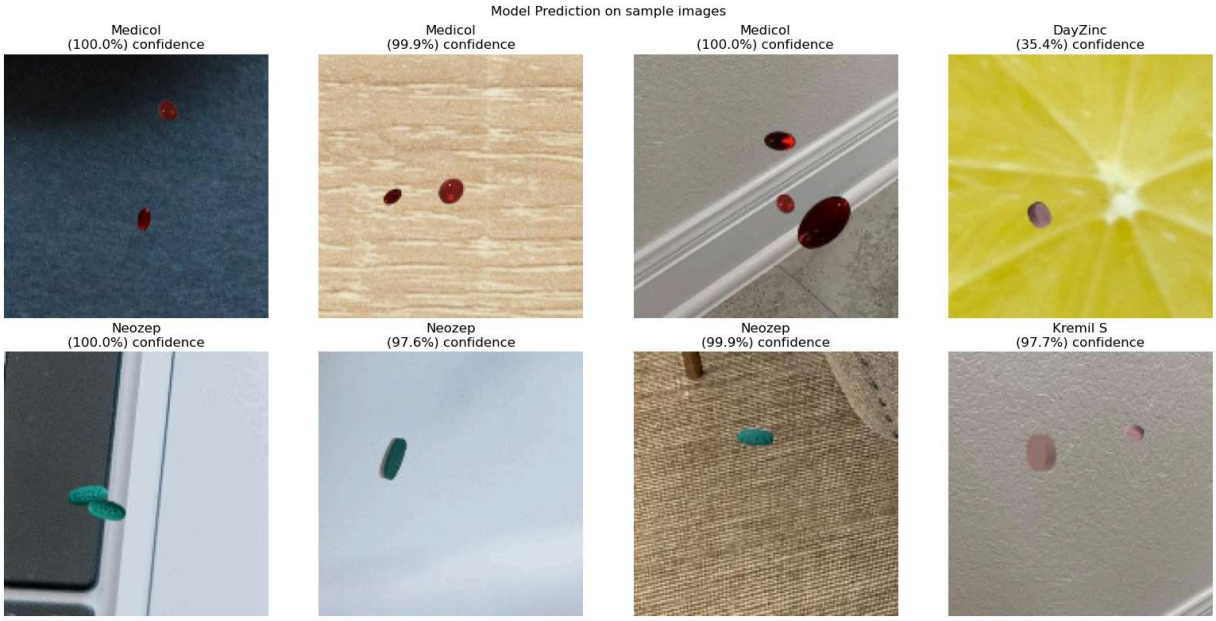


In [34]:
```
train_generator.class_indices
```

Out[34]:
```
{'Alaxan': 0,
 'Bactidol': 1,
 'Bioflu': 2,
 'Biogesic': 3,
 'DayZinc': 4,
 'Decolgen': 5,
 'FishOil': 6,
 'KremilS': 7,
 'Medicol': 8,
 'Neozep': 9}
```

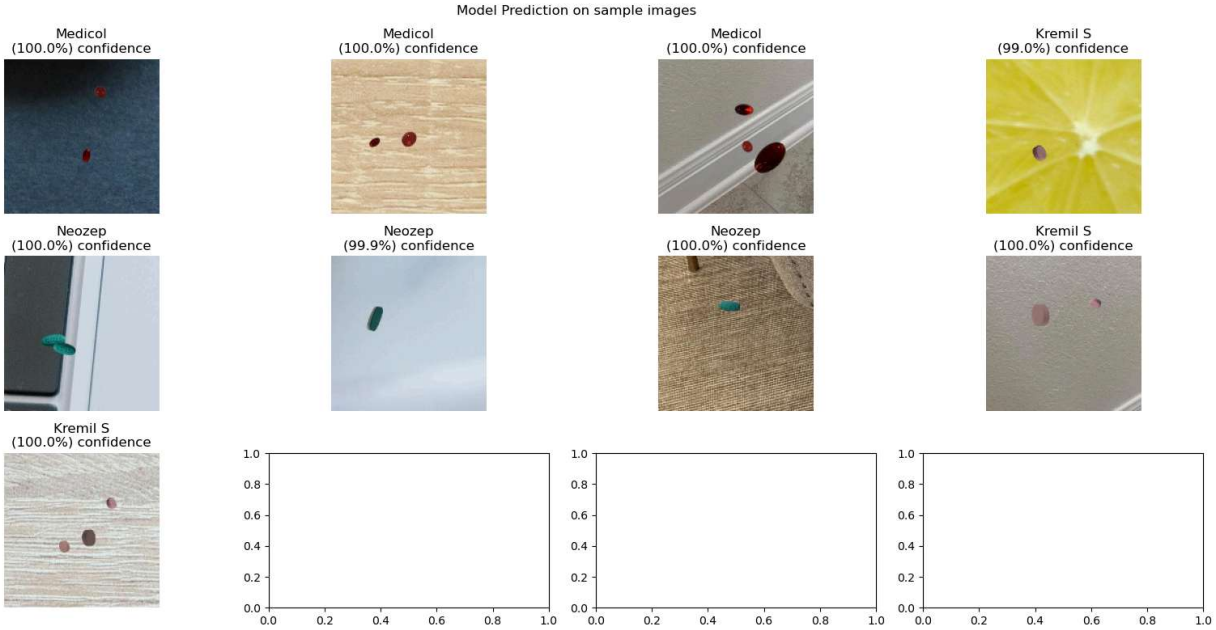
In [35]:
```
# used categorical instead of sparse (1st try)
sample_images = [f for f in os.listdir(sample_dir) if f.endswith(('.jpg','.jpeg','.

fig,axes = plt.subplots(2,4,figsize=(16,8))
axes = axes.flatten()

for i, img_name in enumerate(sample_images[:8]):
    img_path = os.path.join(sample_dir,img_name)
    img = image.load_img(img_path, target_size=(150,150))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array,axis=0)
    img_array = img_array/255.0

    class_names = [
        'Alaxan', 'Bactidol', 'Bioflu', 'Biogesic', 'DayZinc', 'Decolgen', 'Fish Oi
        'Kremil S', 'Medicol', 'Neozep'
    ]

    pred = model.predict(img_array)

    class_index = np.argmax(pred[0]) # ai-generated
    confidence = pred[0][class_index] * 100 # ai-generated
    label = class_names[class_index]

    axes[i].imshow(img)
    axes[i].set_title(f'{label}\n({confidence:.1f}%) confidence')
    axes[i].axis('off')
plt.suptitle('Model Prediction on sample images')
plt.tight_layout()
plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 741ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 82ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step
```



In [35]:
```
# used categorical instead of sparse (FINAL)
sample_images = [f for f in os.listdir(sample_dir) if f.endswith(('.jpg','.jpeg','.

fig,axes = plt.subplots(3,4,figsize=(16,8))
axes = axes.flatten()

for i, img_name in enumerate(sample_images):
    img_path = os.path.join(sample_dir,img_name)
    img = image.load_img(img_path, target_size=(150,150))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array,axis=0)
    img_array = img_array/255.0

    class_names = [
        'Alaxan', 'Bactidol', 'Bioflu', 'Biogesic', 'DayZinc', 'Decolgen', 'Fish Oi
        'Kremil S', 'Medicol', 'Neozep'
    ]

    pred = model.predict(img_array)

    class_index = np.argmax(pred[0]) # ai-generated
    confidence = pred[0][class_index] * 100 # ai-generated
    label = class_names[class_index]

    axes[i].imshow(img)
    axes[i].set_title(f'{label}\n({confidence:.1f}%) confidence')
    axes[i].axis('off')
plt.suptitle('Model Prediction on sample images')
plt.tight_layout()
plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 474ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
```



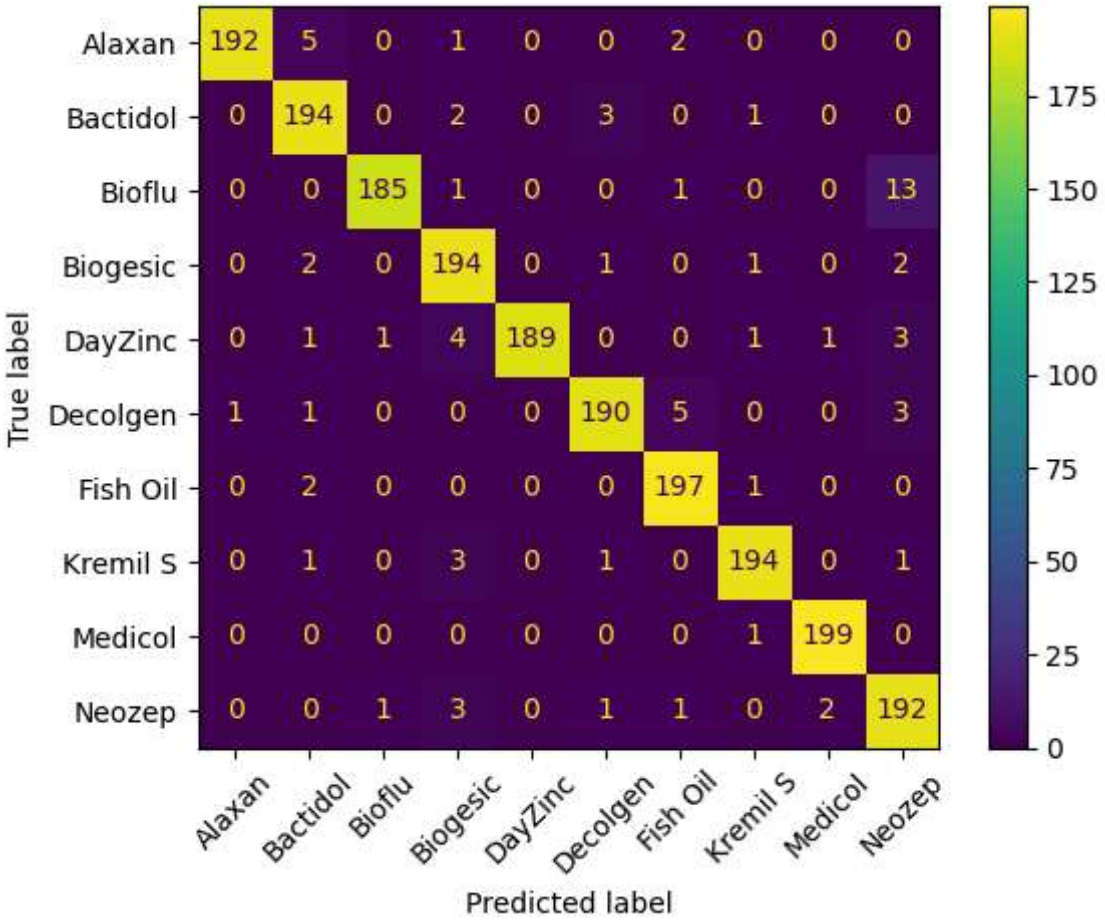
In [36]:
```
y_pred = model.predict(val_generator)
```

```
63/63 ━━━━━━━━━━━━━━━━━━━━ 9s 133ms/step
```

In [37]:
```
val_generator.classes
```

Out[37]:
```
array([0, 0, 0, ..., 9, 9, 9])
```

In [38]:
```
true_classes = val_generator.classes
pred_classes = np.argmax(y_pred, axis=1)
class_names = [
    'Alaxan', 'Bactidol', 'Bioflu', 'Biogesic', 'DayZinc', 'Decolgen', 'Fish Oil',
    'Kremil S', 'Medicol', 'Neozep'
]

ConfusionMatrixDisplay.from_predictions(true_classes,pred_classes,display_labels=cl
plt.xticks(rotation=45)
plt.show()
```



In [32]:
```
# accuracy score
model = tf.keras.models.load_model('pill.keras')
t_lost,t_acc = model.evaluate(val_generator, verbose=1)
```

```
63/63 ━━━━━━━━━━━━━━━━━━━━ 10s 146ms/step - accuracy: 0.9630 - loss: 0.1358
```

In [33]:
```
t_acc
```

Out[33]:
```
0.9629999995231628
```

In [40]:
```
! pip install streamlit
```

```
Requirement already satisfied: streamlit in c:\users\asus\anaconda3\envs\tensorflow_
env\lib\site-packages (1.52.2)
Requirement already satisfied: altair!=5.4.0,!=5.4.1,<7,>=4.0 in c:\users\asus\anaco
nda3\envs\tensorflow_env\lib\site-packages (from streamlit) (6.0.0)
Requirement already satisfied: blinker<2,>=1.5.0 in c:\users\asus\anaconda3\envs\ten
sorflow_env\lib\site-packages (from streamlit) (1.9.0)
Requirement already satisfied: cachetools<7,>=4.0 in c:\users\asus\anaconda3\envs\te
nsorflow_env\lib\site-packages (from streamlit) (6.2.4)
Requirement already satisfied: click<9,>=7.0 in c:\users\asus\anaconda3\envs\tensorf
low_env\lib\site-packages (from streamlit) (8.3.1)
Requirement already satisfied: numpy<3,>=1.23 in c:\users\asus\anaconda3\envs\tensor
flow_env\lib\site-packages (from streamlit) (1.26.4)
Requirement already satisfied: packaging>=20 in c:\users\asus\anaconda3\envs\tensorf
low_env\lib\site-packages (from streamlit) (25.0)
Requirement already satisfied: pandas<3,>=1.4.0 in c:\users\asus\anaconda3\envs\tens
orflow_env\lib\site-packages (from streamlit) (2.3.3)
Requirement already satisfied: pillow<13,>=7.1.0 in c:\users\asus\anaconda3\envs\ten
sorflow_env\lib\site-packages (from streamlit) (12.0.0)
Requirement already satisfied: protobuf<7,>=3.20 in c:\users\asus\anaconda3\envs\ten
sorflow_env\lib\site-packages (from streamlit) (4.25.8)
Requirement already satisfied: pyarrow>=7.0 in c:\users\asus\anaconda3\envs\tensorfl
ow_env\lib\site-packages (from streamlit) (22.0.0)
Requirement already satisfied: requests<3,>=2.27 in c:\users\asus\anaconda3\envs\ten
sorflow_env\lib\site-packages (from streamlit) (2.32.5)
Requirement already satisfied: tenacity<10,>=8.1.0 in c:\users\asus\anaconda3\envs\t
ensorflow_env\lib\site-packages (from streamlit) (9.1.2)
Requirement already satisfied: toml<2,>=0.10.1 in c:\users\asus\anaconda3\envs\tenso
rflow_env\lib\site-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.4.0 in c:\users\asus\anaconda
3\envs\tensorflow_env\lib\site-packages (from streamlit) (4.15.0)
Requirement already satisfied: watchdog<7,>=2.1.5 in c:\users\asus\anaconda3\envs\te
nsorflow_env\lib\site-packages (from streamlit) (6.0.0)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in c:\users\asus\anacond
a3\envs\tensorflow_env\lib\site-packages (from streamlit) (3.1.45)
Requirement already satisfied: pydeck<1,>=0.8.0b4 in c:\users\asus\anaconda3\envs\te
nsorflow_env\lib\site-packages (from streamlit) (0.9.1)
Requirement already satisfied: tornado!=6.5.0,<7,>=6.0.3 in c:\users\asus\anaconda3
\envs\tensorflow_env\lib\site-packages (from streamlit) (6.5.4)
Requirement already satisfied: jinja2 in c:\users\asus\anaconda3\envs\tensorflow_env
\lib\site-packages (from altair!=5.4.0,!=5.4.1,<7,>=4.0->streamlit) (3.1.6)
Requirement already satisfied: jsonschema>=3.0 in c:\users\asus\anaconda3\envs\tenso
rflow_env\lib\site-packages (from altair!=5.4.0,!=5.4.1,<7,>=4.0->streamlit) (4.25.
1)
Requirement already satisfied: narwhals>=1.27.1 in c:\users\asus\anaconda3\envs\tens
orflow_env\lib\site-packages (from altair!=5.4.0,!=5.4.1,<7,>=4.0->streamlit) (2.14.
0)
Requirement already satisfied: colorama in c:\users\asus\anaconda3\envs\tensorflow_e
nv\lib\site-packages (from click<9,>=7.0->streamlit) (0.4.6)
Requirement already satisfied: gitdb<5,>=4.0.1 in c:\users\asus\anaconda3\envs\tenso
rflow_env\lib\site-packages (from gitpython!=3.1.19,<4,>=3.0.7->streamlit) (4.0.12)
Requirement already satisfied: smmap<6,>=3.0.1 in c:\users\asus\anaconda3\envs\tenso
rflow_env\lib\site-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->str
eamlit) (5.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\asus\anaconda3\env
s\tensorflow_env\lib\site-packages (from pandas<3,>=1.4.0->streamlit) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\asus\anaconda3\envs\tensorfl
```

```
ow_env\lib\site-packages (from pandas<3,>=1.4.0->streamlit) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\asus\anaconda3\envs\tensor
flow_env\lib\site-packages (from pandas<3,>=1.4.0->streamlit) (2025.3)
Requirement already satisfied: charset_normalizer<4,>=2 in c:\users\asus\anaconda3\e
nvs\tensorflow_env\lib\site-packages (from requests<3,>=2.27->streamlit) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\asus\anaconda3\envs\tensorfl
ow_env\lib\site-packages (from requests<3,>=2.27->streamlit) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\asus\anaconda3\envs\te
nsorflow_env\lib\site-packages (from requests<3,>=2.27->streamlit) (2.6.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\asus\anaconda3\envs\te
nsorflow_env\lib\site-packages (from requests<3,>=2.27->streamlit) (2025.11.12)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\asus\anaconda3\envs\tenso
rflow_env\lib\site-packages (from jinja2->altair!=5.4.0,!=5.4.1,<7,>=4.0->streamlit)
(3.0.3)
Requirement already satisfied: attrs>=22.2.0 in c:\users\asus\anaconda3\envs\tensorf
low_env\lib\site-packages (from jsonschema>=3.0->altair!=5.4.0,!=5.4.1,<7,>=4.0->str
eamlit) (25.4.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\asus
\anaconda3\envs\tensorflow_env\lib\site-packages (from jsonschema>=3.0->altair!=5.4.
0,!=5.4.1,<7,>=4.0->streamlit) (2025.9.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\asus\anaconda3\envs\t
ensorflow_env\lib\site-packages (from jsonschema>=3.0->altair!=5.4.0,!=5.4.1,<7,>=4.
0->streamlit) (0.37.0)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\asus\anaconda3\envs\tensor
flow_env\lib\site-packages (from jsonschema>=3.0->altair!=5.4.0,!=5.4.1,<7,>=4.0->st
reamlit) (0.30.0)
Requirement already satisfied: six>=1.5 in c:\users\asus\anaconda3\envs\tensorflow_e
nv\lib\site-packages (from python-dateutil>=2.8.2->pandas<3,>=1.4.0->streamlit) (1.1
7.0)
```

In [ ]: