UNIVERSIDAD PERUANA
CAYETANO HEREDIA

Optimized U-Net for Brain Tumor Segmentation

Aplicaciones Clínicas en Señales e Imágenes

Grupo 1

Semestre 2024-2

Contenido

Avance 2 | Preparación de la data

05 Preprocesamiento

Preprocesamiento

1. **Carga de librerías**

```
[ ]  import numpy as np
     import matplotlib.pyplot as plt
     import nibabel as nib
     import json
     import os
     from glob import glob
     from subprocess import call
     import time
     from joblib import Parallel, delayed
```

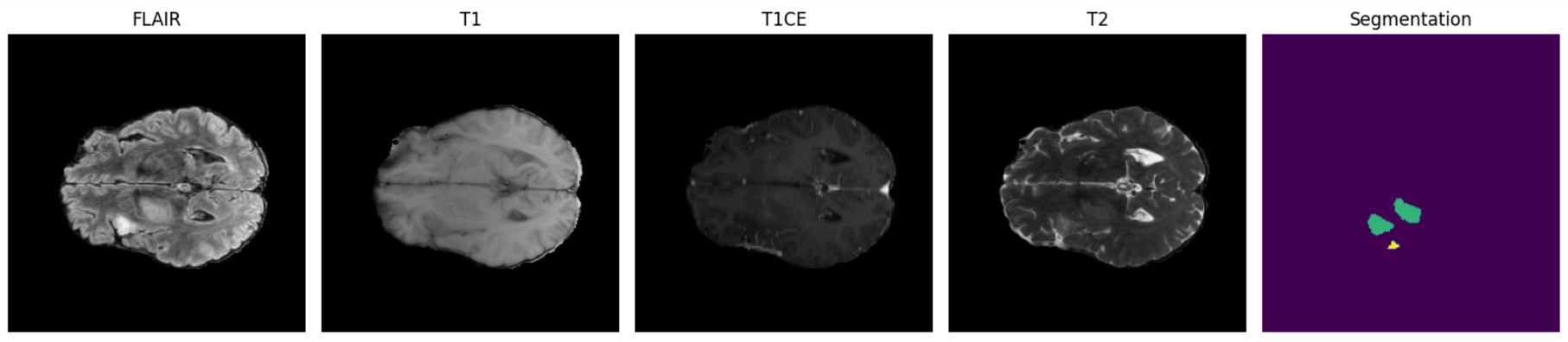
‘**nibabel’:** Para manejar archivos de imágenes médicas (NIfTI)

2. Visualización Inicial de Imágenes

```
[ ]  imgs = [nib.load(f"/content/drive/My Drive/BraTS2021_Training_Data/BraTS2021_00002/BraTS2021_00002_{m}.nii.gz").get_fdata().astype(np.float32)[:, :, 75] for m i
     lbl = nib.load("/content/drive/My Drive/BraTS2021_Training_Data/BraTS2021_00002/BraTS2021_00002_seg.nii.gz").get_fdata().astype(np.uint8)[:, :, 75]

[ ]  fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15, 15))
     for i, img in enumerate(imgs):
         ax[i].imshow(img, cmap='gray')
         ax[i].axis('off')
     ax[-1].imshow(lbl, vmin=0, vmax=4)
     ax[-1].axis('off')
     plt.tight_layout()
     plt.show()
```

Preprocesamiento

2. Visualización Inicial de Imágenes



3. Funciones para Cargar y Procesar Archivos NIfTI

```python
# Cargar el archivo NIfTI desde el directorio
def load_nifty(directory, example_id, suffix):
    return nib.load(os.path.join(directory, example_id + "_" + suffix + ".nii.gz"))

# Cargar todos los canales (flair, t1, t1ce, t2)
def load_channels(d, example_id):
    return [load_nifty(d, example_id, suffix) for suffix in ["flair", "t1", "t1ce", "t2"]]

# Obtener datos en un formato específico
def get_data(nifty, dtype="int16"):
    if dtype == "int16":
        data = np.abs(nifty.get_fdata().astype(np.int16))
        data[data == -32768] = 0
        return data
    return nifty.get_fdata().astype(np.uint8)
```

- **'load_nifty'**: para cargar una imagen específica (paciente o estudio particular)
- **'load_channels'**: carga todas las modalidades en BraTS21
- **'get_data'**: Extrae los datos volumétricos (3D) de u objeto NIfTI cargado y realiza limpieza de valores atípicos o inválidos

Preprocesamiento

4. Preparación y Guardado de Volúmenes

```python
def prepare_nifty(d, img_path, lbl_path):
    example_id = d.split("/")[-1]
    flair, t1, t1ce, t2 = load_channels(d, example_id)
    affine, header = flair.affine, flair.header
    vol = np.stack([get_data(flair), get_data(t1), get_data(t1ce), get_data(t2)], axis=-1)
    vol = nib.nifti1.Nifti1Image(vol, affine, header=header)
    nib.save(vol, os.path.join(img_path, example_id + ".nii.gz"))

    if os.path.exists(os.path.join(d, example_id + "_seg.nii.gz")):
        seg = load_nifty(d, example_id, "seg")
        affine, header = seg.affine, seg.header
        vol = get_data(seg, "unit8")
        vol[vol == 4] = 3
        seg = nib.nifti1.Nifti1Image(vol, affine, header=header)
        nib.save(seg, os.path.join(lbl_path, example_id + "_seg.nii.gz"))
```

- **'load_nifty'**: para cargar una imagen específica (paciente o estudio particular)
- **'load_channels'**: carga todas las modalidades en BraTS21
- **'get_data'**: Extrae los datos volumétricos (3D) de u objeto NIfTI cargado y realiza limpieza de valores atípicos o inválidos

5. Preparación del Conjunto Completo de Datos

```python
def prepare_dataset(data):
    img_path = os.path.join(data, "images")
    lbl_path = os.path.join(data, "labels")
    print(f"Preparing BraTS21 dataset from: {data}")
    start = time.time()
    dirs = sorted(glob(os.path.join(data, "BraTS*")))
    run_parallel(prepare_nifty, [(d, img_path, lbl_path) for d in dirs])
    end = time.time()
    print(f"Preparing time: {(end - start):.2f}")
```

- **'load_nifty'**: para cargar una imagen específica (paciente o estudio particular)
- **'load_channels'**: carga todas las modalidades en BraTS21
- **'get_data'**: Extrae los datos volumétricos (3D) de u objeto NIfTI cargado y realiza limpieza de valores atípicos o inválidos

Preprocesamiento

6. Recorte y Normalización de Volúmenes

```python
def normalize(volume):
    """Normalizar el volumen para que los valores estén entre 0 y 1."""
    volume = (volume - np.min(volume)) / (np.max(volume) - np.min(volume))
    return volume

def crop_center(volume, mask):
    """Recortar el volumen y la máscara centrados en la región no nula."""
    non_zero_coords = np.where(mask > 0)
    min_z, max_z = np.min(non_zero_coords[2]), np.max(non_zero_coords[2])
    min_y, max_y = np.min(non_zero_coords[1]), np.max(non_zero_coords[1])
    min_x, max_x = np.min(non_zero_coords[0]), np.max(non_zero_coords[0])

    cropped_volume = volume[min_x:max_x, min_y:max_y, min_z:max_z]
    cropped_mask = mask[min_x:max_x, min_y:max_y, min_z:max_z]
    return cropped_volume, cropped_mask
```

Normaliza los valores del volumen a 0 y 1 y recorta el volumen y la máscara centrados en la región que contiene datos no nulos (la región del tumor), eliminando áreas irrelevantes.

7. Preprocesamiento y Guardado de Volúmenes Normalizados y Recortados

```python
def preprocess_and_save(data_dir, img_save_dir, lbl_save_dir):
    """Preprocesar los volúmenes y guardarlos como matrices NumPy en carpetas separadas."""
    os.makedirs(img_save_dir, exist_ok=True)
    os.makedirs(lbl_save_dir, exist_ok=True)

    img_files = sorted(glob(os.path.join(data_dir, "images", "*.nii.gz")))
    lbl_files = sorted(glob(os.path.join(data_dir, "labels", "*.nii.gz")))

    for img_file, lbl_file in zip(img_files, lbl_files):
        # Cargar imágenes y etiquetas
        img = nib.load(img_file).get_fdata().astype(np.float32)
        lbl = nib.load(lbl_file).get_fdata().astype(np.uint8)

        # Recortar y normalizar las imágenes y etiquetas
        img_cropped, lbl_cropped = crop_center(img, lbl)
        img_normalized = normalize(img_cropped)

        # Guardar las imágenes preprocesadas
        img_filename = os.path.basename(img_file).replace('.nii.gz', '_preprocessed.npy')
        np.save(os.path.join(img_save_dir, img_filename), img_normalized)

        # Guardar las etiquetas preprocesadas
        lbl_filename = os.path.basename(lbl_file).replace('.nii.gz', '_preprocessed.npy')
        np.save(os.path.join(lbl_save_dir, lbl_filename), lbl_cropped)
```

Preprocesar los volúmenes de imágenes y las segmentaciones recortando y normalizando, y luego guardarlos en formato NumPy para un acceso más rápido y eficiente durante el entrenamiento del modelo.

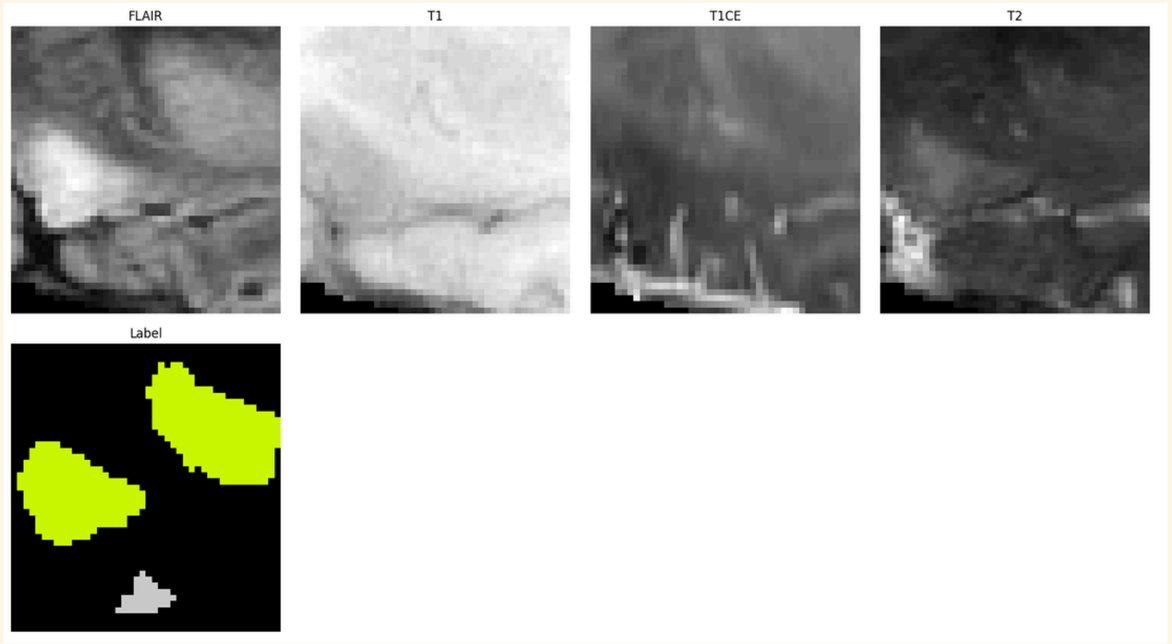
Preprocesamiento

8. Visualización de Imágenes Preprocesadas

```
titles = ['FLAIR', 'T1', 'T1CE', 'T2', 'Label']

# Graficar las imágenes de diferentes modalidades
for i in range(4):
    ax[0, i].imshow(img[:, :, z_slice, i], cmap='gray')
    ax[0, i].set_title(titles[i])
    ax[0, i].axis('off')

# Graficar la etiqueta
ax[1, 0].imshow(lbl[:, :, z_slice], cmap='nipy_spectral')
ax[1, 0].set_title('Label')
ax[1, 0].axis('off')

# Eliminar los ejes no utilizados
for i in range(1, 5):
    ax[0, i].axis('off')
for i in range(1, 5):
    ax[1, i].axis('off')

# Ajustar el espaciado
plt.tight_layout(pad=2.0)
plt.show()
```

ostrando las diferentes modalidades de imagen junto con la segmentación en un corte específico, para verificar visualmente el resultado del preprocesamiento.

Preprocesamiento



Preprocesamiento final

GRACIAS

Provide a brief **overview** of the pitch deck's content

Engage the audience with a compelling **introduction**

Identify the customer's pain points and **challenges**

Describe how your product or service can **solve the problem**

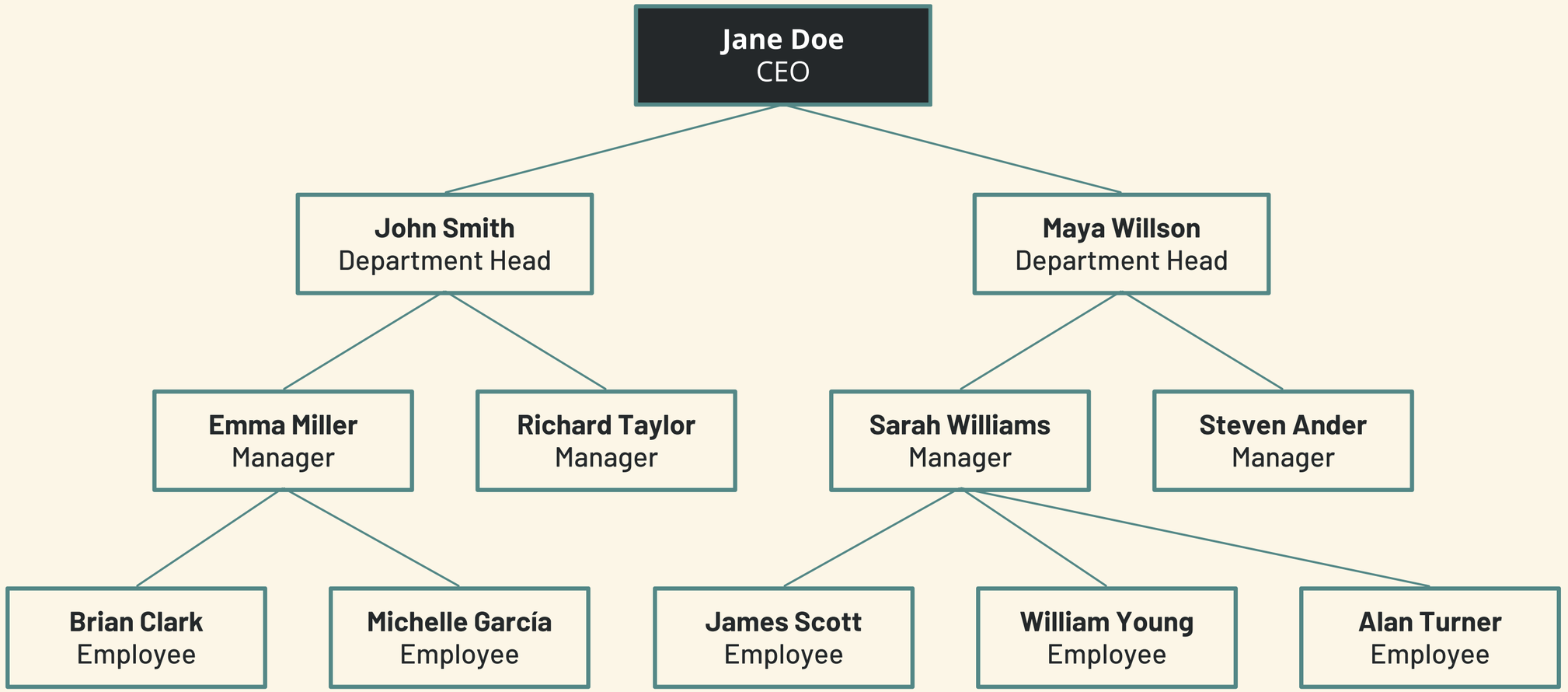
Highlight the unique value proposition and **benefits** of your specific solution

Analyze the target market size, growth potential, and competition briefly

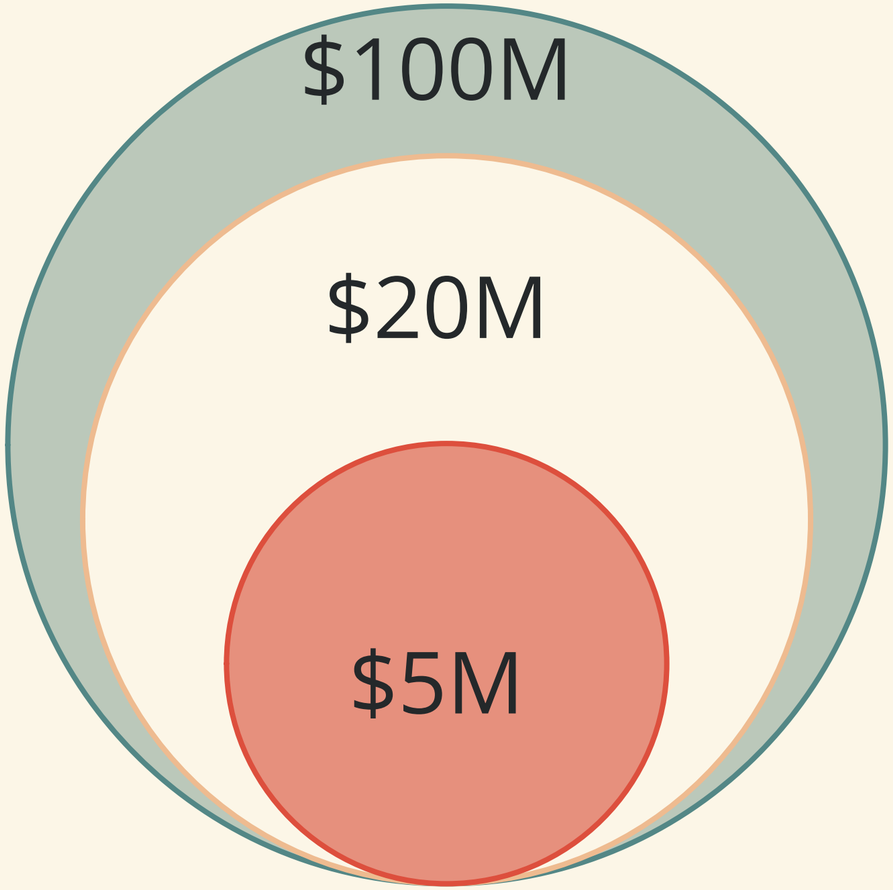
Explain your financial needs and briefly outline your **funding allocation**

End with a clear and concise **call to action**

Organizational chart



Market size overview



Outer circle

Include the total size of the market, which represents the entire potential customer base for the product or service

Middle circle

Identify the target market for the product or service, which may be a subset of the total market. This could be based on factors such as demographics, geography, or specific needs

Inner circle

Indicate the current market size, which represents the portion of the target market that the company has successfully captured

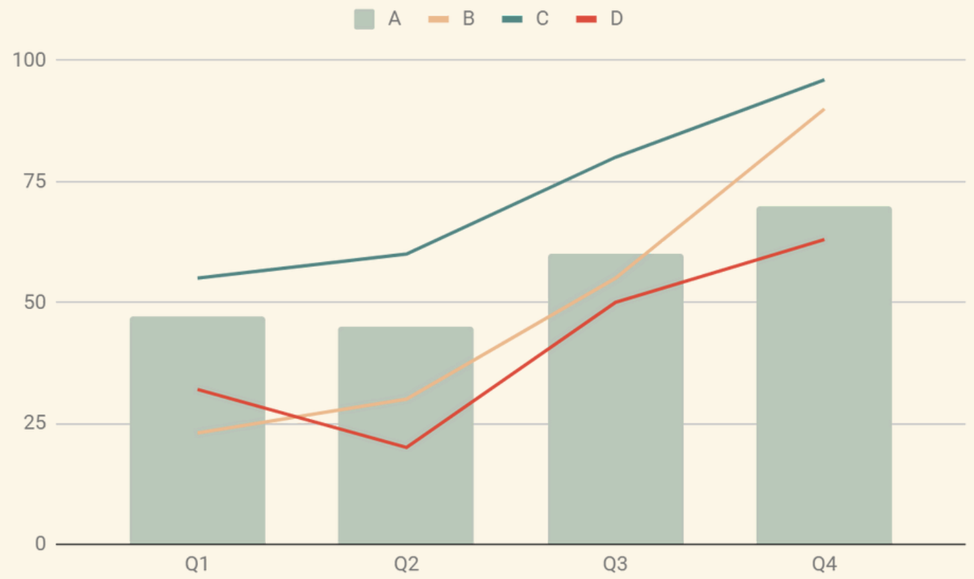
Roadmap infographic

Initiative	Objective	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Understanding	Analyze and understand the needs of your target audience												
Conduct research	Research existing products in the industry and analyze how successful they are												
Brainstorm ideas	Generate ideas based on user feedback and research findings												
Develop a prototype	Create a basic version of the product to show investors												
Test for usability	Put the prototype through rigorous testing processes to ensure that it meets user requirements												
Analyze feedback	Understand the opinion of the users who tried your product												

KPI dashboard

Product	Units	Revenue	Returns
Item 1	500	2,000,000	40
Item 2	1,000	50,750	10
Item 3	250	1,500,000	300
Item 4	500	2,000,000	40
Item 5	1,000	50,750	10
Item 6	250	1,500,000	300
Item 7	500	2,000,000	40
Item 8	1,000	50,750	10

$4,000,000	$100,000	$5,000
Revenue	General costs	Profit



Follow the link in the graph to modify its data and then paste the new one here. **For more info, click here**

Thanks!

Do you have any questions?

youremail@freepik.com
+34 654 321 432
yourwebsite.com

Icon pack