O código apresentado é uma implementação de um detector de bolas natalinas em imagens.

As principais funções utilizadas no código são as seguintes:

- **detect_edges(image, threshold)**:
    - Essa função utiliza o operador de Sobel para detectar as bordas de uma imagem. Ela retorna uma imagem binária com as bordas marcadas. O **operador de Sobel** é amplamente utilizado devido à sua simplicidade computacional e eficácia na detecção de bordas em diferentes orientações. Ele é especialmente útil para detectar bordas em imagens com ruído ou com variações de intensidade de fundo, o que o torna uma escolha comum em muitos algoritmos de detecção de bordas em processamento de imagem.
- **make_annulus_kernel(outer_radius, annulus_width)**
    - Essa função cria um kernel em forma de anel para ser utilizado na detecção de círculos. O kernel é uma matriz de números, que será convoluída com a imagem original a fim de identificar regiões que se assemelham a círculos.
- **detect_circles(image, radii, annulus_width)**
    - Essa função realiza a convolução da imagem com o kernel em forma de anel para detectar os círculos presentes na imagem. Na convolução no domínio do espaço cada pixel seria percorrido e convolvido com o kernel gerado na função **make_annulus_kernel()**, a partir disso seria gerada uma matriz de acumulação para cada pixel, onde o pixel de centro com maior semelhança com anel gerado teria maior sinal. A convolução utilizando a transformada rápida de Fourier vai facilitar o processo, já que uma convolução no domínio do tempo se trata de uma multiplicação no domínio da frequência. A função retorna um array com as coordenadas dos círculos detectados e seus respectivos raios.
- **display_results(image, edges, center, radius)**
    - Essa função exibe as bordas da imagem original e marca o círculo detectado com um círculo vermelho. Ela retorna a figura resultante.
- **top_n_circles(acc, radii, n)**
    - Essa função seleciona o círculo com o maior sinal dentre os círculos detectados. Ela retorna as coordenadas do centro do círculo e seu raio.
- **detect_circle_from_file(image)**
    - Essa função recebe uma imagem como entrada e utiliza as funções de detecção de bordas e círculos para retornar a figura resultante com o círculo detectado marcado.
- **detect_circle(image, preprocess=False)**

- Essa função recebe uma imagem e uma flag para determinar se a imagem deve ser pré-processada ou não. Se a flag for verdadeira, a imagem é convertida para escala de cinza e suavizada com um filtro Gaussiano **(Isso é feito para reduzir o ruído presente na imagem e suavizar as transições de intensidade, o que pode melhorar a precisão e a robustez da detecção de bordas.)**. Ela retorna a figura resultante com o círculo detectado marcado.

O fluxograma do código é basicamente a seguinte sequência de passos:

1. Criação de elementos da interface gráfica com a biblioteca Streamlit;
2. Definição dos parâmetros de detecção de círculos (raio máximo e mínimo);
3. Carregamento da imagem pelo usuário;
4. Pré-processamento da imagem;
5. Detecção das bordas da imagem utilizando a função **detect_edges()**;
6. Detecção dos círculos presentes na imagem utilizando a função **detect_circles()**;
7. Seleção do círculo com o maior sinal dentre os círculos detectados utilizando a função **top_n_circles()**;
8. Exibição da figura resultante com o círculo detectado marcado utilizando a função **display_results()**.